Avances en Informática y Automática

Décimo Workshop

VNiVERSiDAD D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Avances en Informática y Automática

Décimo Workshop

Avances en Informática y Automática

Décimo Workshop

Editores

Alejandro Benito Santos
Daniel López Sánchez

Publicado en España por:

Departamento de Informática y Automática Facultad de Ciencias
Universidad de Salamanca
Plaza de los Caídos s/n
37008, Salamanca, España
Tel.: + 34 923294653
Fax: + 34 923294514
Web: `http://mastersi.usal.es`
Email: `mastersi@usal.es`

ISBN **978-84-617-9707-3**

Editores:

Alejandro Benito Santos
Daniel López Sánchez

Prólogo

El Máster Oficial en Sistemas Inteligentes de la Universidad de Salamanca tiene como principal objetivo promover la iniciación de los estudiantes en el ámbito de la investigación. El congreso organizado por el Departamento de Informática y Automática que se celebra dentro del Máster en Sistemas Inteligentes de la Universidad de Salamanca proporciona la oportunidad ideal para que sus estudiantes presenten los principales resultados de sus Trabajos de Fin de Máster y obtengan una realimentación del interés de los mismos.

La décima edición del *workshop* "Avances en Informática y Automática", correspondiente al curso 2015 - 2016, ha sido un encuentro interdisciplinar donde se han presentado trabajos pertenecientes a un amplio abanico de líneas de investigación, desde los sistemas biométricos y la visualización de la información hasta la minería de datos pasando por otros campos relacionados. Todos los trabajos han sido supervisados por investigadores de reconocido prestigio pertenecientes a la Universidad de Salamanca, proporcionando el marco idóneo para sentar las bases de una futura tesis doctoral. Entre los principales objetivos del congreso se encuentran:

- Ofrecer a los estudiantes un marco donde exponer sus primeros trabajos de investigación.
- Proporcionar a los participantes un foro donde discutir ideas y encontrar nuevas sugerencias de compañeros, investigadores y otros asistentes a la reunión.
- Permitir a cada estudiante una realimentación de los participantes sobre su trabajo y una orientación sobre las futuras direcciones de investigación.
- Contribuir al desarrollo del espíritu de colaboración en la investigación.

Organización

El *workshop* "Avances en Informática y Automática" está organizado por el Departamento de Informática y Automática de la Universidad de Salamanca.

Comité Organizador

Dr. Francisco J. Blanco Rodríguez
Dr. Emilio S. Corchado Rodríguez
Dra. Belén Curto Diego
Dr. José Rafael García-Bermejo Giner
Dra. Vivian F. López Batista
Dra. María Gracia Manzano Arjona
Dr. Vidal Moreno Rodilla
Dr. Roberto Therón Sánchez
Dra. Pastora I. Vega Cruz

Índice general

Generación automática de listas de reproducción de música mediante técnicas de minería de datos

María Arista y María Moreno

Universidad de Salamanca
Departamento de Informática y Automática
Facultad de Ciencias
Plaza de los Caídos s/n
37008 Salamanca, Spain

maria.arista@usal.es, mmg@usal.es,
http://www.usal.es

Resumen Los sistemas de recomendación representan un medio eficaz de filtrado cuando existe mucha información disponible para el usuario. Los métodos de recomendación más conocidos y ampliamente desarrollados en los últimos años son los de filtrado colaborativo, en los que se utilizan las preferencias de los usuarios para encontrar otros con preferencias similares para predecir los ítems en los que un usuario podría estar interesado. Este trabajo se enfoca en los sistemas de recomendación de música, ya que debido al aumento de consumo musical, estos sistemas pueden resultar útiles para la creación de listas de reproducción de música relevante y novedosa en función de los gustos musicales personales del usuario. En este sentido, se ha desarrollado un método híbrido de recomendación, que integra las técnicas de filtrado colaborativo y las basadas en contenido. Los resultados revelaron que al aplicar conjuntamente los métodos de filtrado colaborativo y basado en contenido, la fiabilidad de las recomendaciones es mayor, que al utilizarlos de forma independiente.

Keywords: Filtrado colaborativo, basado en contenido, recomendación de música, híbrido, API Last.fm

1. Introducción

En los últimos años, la web se ha convertido en la principal fuente de títulos de música en formato digital. Los millones de pistas disponibles y la variedad de sitios web hace que la búsqueda de las canciones constituya un problema para los usuarios. Ésta es la razón del gran interés en el desarrollo de algoritmos de recomendación que permitan la generación automática de listas personalizadas de reproducción. El propósito de este trabajo se centra en la generación de listas de reproducción personalizadas mediante la aplicación de técnicas de filtrado colaborativo. Para ello, se tendrá en cuenta el perfil de escucha de los usuarios y la similitud de preferencias con otros usuarios del sistema (cuyos datos se han recopilado desde la API de *Last.fm*. Los métodos de filtrado colaborativo

requieren las valoraciones del usuario sobre los productos a recomendar, en este caso las canciones de las listas de reproducción. Sin embargo, en las bases de datos sobre música no se dispone de valoraciones explícitas de los usuarios por lo que ha sido necesario recurrir a un tratamiento de los datos para obtener de forma implícita las preferencias de los usuarios.

2. Sistemas de Recomendación

Los sistemas de recomendación son un tipo específico de filtro de información cuyo objetivo es mostrar ítems al usuario que le sean relevantes o de interés. Se entiende por filtro de información, a un sistema que elimina información no deseada de un flujo de información de forma automática o semiautomática para ser presentada a los usuarios [1]. A continuación se mencionan las principales categorías de sistemas de recomendación [7]:

- **Recomendación Colaborativa**: Se puede resumir en la frase *"Enséñame lo que es popular entre mis vecinos"*. También llamado Filtrado colaborativo (*Collaborative filtering*), mediante el cual se recomiendan los ítems que les gustan a los usuarios similares ("vecinos") al usuario activo. Se asume que si los usuarios han compartido algunos de sus intereses en el pasado, tendrán gustos similares en el futuro. La entrada que recibe este sistema es una matriz de usuarios-ítems, y el resultado va a depender del objetivo buscado. Puede devolver la predicción del *rating* (valoración) de cada ítem o puede devolver una lista de N ítems recomendados que no han sido vistos aún por el usuario activo.
- **Recomendación basada en el contenido (*Content based*)**: Se puede describir mediante la frase *"Muéstrame más de lo que ya me ha gustado"*. En estos sistemas los perfiles de usuario se construyen a partir de las características de los ítems que un usuario ha valorado muy positivamente. Se emparejan aquellos ítems que mejor cumplan las preferencias del usuario y que aún no han sido probados por él. La tarea de esta técnica es la de aprender las preferencias del usuario y recomendar aquellos elemenos que son similares a sus gustos.
- **Recomendación basada en el conocimiento (*Knowledge based*)**: *"Enséñame lo que se adapta a mis necesidades"* es la frase que se ajusta a esta categorías de métodos. Esta técnica se usa cuando no se tiene historial en el caso de los usuarios nuevos o cuando los ítems tienen un bajo número de valoraciones por ser productos que se han introducido recientemente en el sistema. Son recomendadores que están más centrados en el dominio de la aplicación.
- **Recomendación híbrida**: Esta categoría combina dos o más tipos de las técnicas anteriormente mencionadas. Su propósito es aprovechar las ventajas y evitar los incovenientes de dichos métodos para obtener "mejores" recomendaciones.

2.1. Métodos de Filtrado Colaborativo

Desde el punto de vista de Sarwar et al. [12], el filtrado colaborativo considera un conjunto de m usuarios U $= \{u_1, u_2, ..., u_m\}$, y otro de n ítems I $= \{i_1, i_2, ..., i_n\}$. Además, cada usuario u_i posee una lista de k valoraciones de un conjunto de ítems I_{u_i} que ha realizado, donde $I_{u_i} \subseteq$ I. En ese sentido, una recomendación realizada, al usuario activo $u_a \in$ U, consiste en un conjunto de N ítems $I_r \subset$ I a los cuales se predijo que el usuario activo tendría interés. Cabe resaltar que $I_r \cap I_{u_a} = \emptyset$, pues se desea que se recomienden ítems que el usuario no haya consumido aún. Las valoraciones se almacenan en una matriz de valoraciones donde se verifica si el usuario u_a podrá tener interés por el ítem i_j.

Los algoritmos de recomendación colaborativos pueden ser clasificadas en dos tipos según la forma en que procesen la información [13]:

Algoritmos basados en memoria En estos algoritmos las predicciones son concebidas utilizando todas las valoraciones disponibles en el sistema. Es decir, procesa la matriz de valoraciones cada vez que calcula una predicción. Generalmente, usan medidas de similitud para encontrar usuarios o ítems con un patrón de valoraciones similar, y los utilizan para llevar a cabo la recomendación. Para proporcionar recomendaciones a un usuario, la primera tarea a desempeñar por los algoritmos de esta categoría es la computación de la similitud entre los usuarios del sistema. Para ello, según Sarwar et al. [12], usualmente se utiliza el coeficiente de correlación de Pearson. La ecuación siguiente aclara como tal coeficiente puede ser calculado para dos usuarios, donde $v_{a,j}$ expresa una valoración realizada por el usuario activo acerca de un producto j, $\bar{v}_i$ el promedio de las valoraciones del usuario i y $\bar{v}_a$ el promedio de las valoraciones del usuario activo.

$$w(a,i) = \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{a,j} - \bar{v}_a)^2 \sum_j (v_{i,j} - \bar{v}_i)^2}} \tag{1}$$

Después de disponer de los coeficientes de similitud, el algoritmo obtiene una lista de los n usuarios U $= \{u_1, u_2, ..., u_n\}$ más similares al usuario activo u_a, donde u_1 es el usuario más próximo a u_a, u_2 es el segundo más próximo y así sucesivamente. Una vez obtenidos los vecinos más cercanos a u_a, ya se puede realizar la predicción referente a un determinado ítem j, la cual define si tal ítem va a ser recomendado o no.

Algoritmos basados en modelo En estos algoritmos se utiliza parte de las preferencias para construir un modelo de estimación de valoraciones. Es decir, hacen uso de la información presente en la matriz de valoraciones para entrenar un modelo previamente especificado. Las predicciones se hacen directamente a partir de dicho modelo, sin necesidad de volver a procesar la matriz. Los métodos basados en modelos también son conocidos como métodos "basados en ítems", pues según Sarwar et al. [12], a diferencia de los métodos basados en memoria,

dichos métodos también consideran, para predecir la valoración acerca de un determinado ítem, la similitud de tal ítem con aquellos ítems que el usuario ya haya valorado. Para ello, generalmente se utilizan los mismos tipos de métricas que se utilizan para calcular similitudes. Tratándose de ítems, la correlación de Pearson, por ejemplo, puede ser obtenida, en relación a dos ítems p y q, como a continuación, donde $\bar{v}_p$ y $\bar{v}_q$ expresan, respectivamente, los promedios de valoraciones en relación a los ítems p y q:

$$sim(p,q) = \frac{\sum_{u \in U}(v_{u,p} - \bar{v}_p)(v_{u,q} - \bar{v}_q)}{\sqrt{\sum_{u \in U}(v_{u,p} - \bar{v}_p)^2}\sqrt{\sum_{u \in U}(v_{u,q} - \bar{v}_q)^2}} \tag{2}$$

Otro método para calcular la similitud entre ítems es la medida del coseno [12], donde se considera a cada ítem como un vector dentro de un espacio vectorial de m dimensiones.La similitud entre ellos sería el coseno del ángulo que forman. En otras palabras, en la matriz de valoraciones $m \times n$, la similitud entre los ítems i y j viene dada por:

$$sim(p,q) = \cos(\boldsymbol{i}, \boldsymbol{j}) = \frac{\boldsymbol{i} \cdot \boldsymbol{j}}{\|\boldsymbol{i}\|_2 * \|\boldsymbol{j}\|_2} \tag{3}$$

donde "·" es el producto de los dos vectores.

Después de disponer de los datos relativos a la similitud de los ítems presentes en un sistema, se construye un modelo de estimación de valoraciones, el cual se crea *off-line*, es decir, con anterioridad a la entrada del usuario activo en el sistema. Por lo tanto, se realiza el aprendizaje de un modelo de comportamiento de usuario con datos almacenados por el sistema y de modo *off-line* para que posteriormente tal modelo sea aplicado en tiempo real [8]. Para ello, se suelen utilizar, en general, técnicas aprendizaje automático.

2.2. Trabajos Relacionados

Actualmente muchos investigadores están trabajando en el desarrollo y mejora de métodos de recomendación de música. Algunos trabajos relacionados son como el de Bogdanov et al. [2], que proponen tres enfoques basados en el contenido. Dos de ellos emplean una medida de similitud semántica de la música para generar recomendaciones; y el tercero crea un modelo probabilístico de las preferencias del usuario en el dominio semántico. Otro, el de Park et al. [10], que plantean un método de recomendación llamado *Session-based Collaborative Filtering (SSCF)*, éste incluye los ítems seleccionados actualmente en el perfil de la sesión y encuentra las sesiones más similares para generar la recomendación. Otro trabajo es el de Dias et al. [3] que hicieron uso del contexto temporal y la diversidad de sesiones en las técnicas de filtrado colaborativo basado en la sesión para la recomendación de música. Además, compararon dos algoritmos que utilizan características temporales: uno que extrae de forma explícita las propiedades temporales y la diversidad de sesiones y otro que es capaz de modelar implícitamente patrones temporales. Xing et al. [16] introdujeron el concepto de exploración en el filtrado colaborativo y tratan de compensarlo con la explotación que estos sistemas suelen utilizar. Ellos diseñaron un algoritmo de inferencia

bayesiana para estimar de manera eficiente las distribuciones de probabilidad a posteriori del *rating*. El trabajo de Su et al. [14] propone un sistema de recomendación que utiliza etiquetas de medios sociales para calcular la similitud entre las canciones. Al igual que en el presente trabajo, se usa el número de *plays* de las canciones para calcular las valoraciones, pero de una manera distinta. Hornung et al. [5] presentan un sistema híbrido de recomendación que fusiona tres técnicas de recomendación que son el uso de la similitud de la canción, de la etiqueta y la similitud del tiempo. La nueva música se descubre con el uso de una nueva métrica llamada *serendipia*. Asimismo, Domingues et al. [4] proponen un sistema de recomendación de música híbrido, que combina datos de uso y contenido. Hicieron un estudio comparativo de este sistema híbrido con los sistemas basados en el uso y contenido, de forma independiente. Los resultados mostraron que el primero tenía ventajas sobre los otros. Van den Oord et al. [15] utilizan una matriz de factorización modificada dirigida a conjuntos de datos de valoración implícita, propuesto por Hu et al. [6]. En primer lugar, ellos obtuvieron vectores de variables ocultas de las canciones que utilizaron posteriormente para el entrenamiento de un modelo de regresión. Utilizaron redes neuronales profundas convolucionales para predecir variables ocultas de audio de la música. Y Reafee et al. [11] proponen la inclusión de las relaciones sociales implícitas para la recomendación, para ello proponen un modelo *EISR (explicit and implicit social relation)* y también, un algoritmo *PFNF (Possibility of Friendship Between Non-Friends)* para extraer la relación implícita (relación social escondida) en los grafos no dirigidos de redes sociales mediante la explotación de las técnicas de predicción de enlace. Tanto las amistades explícitas como implícitas se incorporan en el modelo propuesto.

3. Conjunto de Datos "Last.fm"

El proceso de generación del conjunto de datos se ha realizado con el programa desarrollado en *Python* llamado: *generDatosLastFM.py*. Para el almacenamiento de los datos recolectados se ha utilizado la base de datos *Postgresql*. La comunicación entre estas dos tecnologías ha sido posible gracias al uso de la librería *psycopg2* y finalmente, para la comunicación de Python con el API de *Last.fm* se ha usado la librería *PyLast*. Para este fin, el proceso se inicia con el descubrimiento de nombres de usuario de *Last.fm* utilizando el método *User.getFriends*. Para cada uno de ellos, se recupera el país de dónde es. Por cada usuario, un proceso iterativo recupera la totalidad de su perfil de escucha haciendo uso del método *User.getRecentTracks*. Después de recopilar los datos, viene la etapa del filtrado que busca obviar los registros del conjunto de datos que no generen valor.

3.1. Cálculo del Rating

Para el cálculo se ha seguido el método Pacula [9] que se basa en la frecuencia de *plays* donde claramente existe una distribución de ley de potencias, pues hay

pocas canciones escuchadas muy frecuentemente y la mayoría de ellas tienen pocos *plays*. La frecuencia de *plays* de una determinada canción i y un usuario j se define de la siguiente manera:

$$Freq_{i,j} = \frac{p_{i,j}}{\sum_{i'} p_{i',j}} \tag{4}$$

Donde $p_{i,j}$ es el número de veces que un usuario j escucha una canción i. Por otro lado, $Freq_k(j)$ denota la k-ésima canción más escuchada por el usuario j. Entonces, una calificación para una canción con rango k se calcula como una función lineal de la frecuencia percentil:

$$r_{i,j} = 4(1 - \sum_{k'=1}^{k-1} Freq_{k'}(j)) \tag{5}$$

Una vez que el *rating* es calculado, los métodos de filtrado colaborativo pueden ser aplicados tal y como se aplican en un conjunto de datos que contiene las preferencias del usuario de forma explícita. Las características del conjunto de datos final despúes de la etapa de filtrado son:

Cantidad de Usuarios	820
Cantidad de Artistas con Id	2544
Cantidad de Canciones con Id	13888
Cantidad de Registros Totales	22501

Tabla 1: Número de registros del conjunto de datos final

4. Recomendación basada en *Rating*

Este tipo de recomendación no produce una lista de canciones sino que sólo proporciona la predicción del *rating* de las mismas para un usuario determinado. Sin embargo, se pueden utilizar estas predicciones para elaborar listas de reproducción personalizadas para los usuarios que incluyan aquellas canciones que tengan los mejores valores de los *ratings* predichos para cada uno de ellos.

4.1. Algoritmo *User k-NN*

Con este algoritmo se aplica la técnica de filtrado colaborativo de los vecinos más cercanos basado en usuario. Es decir, calcula la similitud mediante la búsqueda de un grupo de k usuarios que han valorado las canciones de una manera muy similar al usuario en cuestión. Para esto se deben establecer dos opciones: el número de vecinos más próximos, k y la medida de similitud. La similitud Coseno es muy popular, sin embargo el coeficiente de correlación de Pearson tiende a ser un poco más precisa. La figura 1 muestra cómo fue implementada la validación de este algoritmo.

Las medidas de calidad de las predicciones, con el valor de k=80, fueron: Usando la similitud Coseno: RMSE = 1.178, MAE = 0.96 y NMAE = 0.24 y el

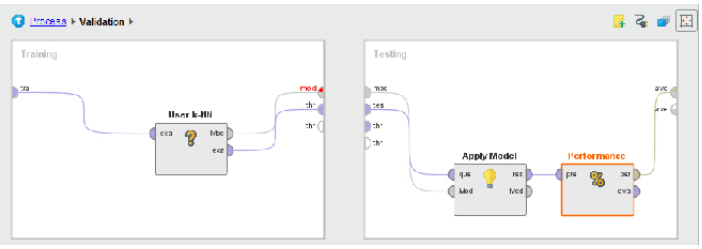


Figura 1: Recomendación basada en Rating: User k-NN

coeficiente de correlación de Pearson: RMSE = 1.15, MAE = 0.938 y NMAE = 0.235.

4.2. Algoritmo *Item k-NN*

Con este algoritmo se aplica la técnica de filtrado colaborativo de los vecinos más cercanos basado en ítems. Es decir que es similar al algoritmo *User k-NN* pero en este caso, la métrica de similitud es aplicada entre los ítems. La manera en cómo se implementó esta validación es similar a la anterior, sólo que para este caso se usa el operador *Item k-NN*. Las medidas de calidad de las predicciones, con el valor de k=80, fueron:

Usando la similitud Coseno: RMSE = 1.077, MAE = 0.875 y NMAE = 0.219 y el coeficiente de correlación de Pearson: RMSE = 1.081, MAE = 0.882 y NMAE = 0.221.

4.3. Algoritmo *User Attribute k-NN*

Con este algoritmo se aplica la técnica de recomendación basada en características del usuario, en este caso se basa en el atributo *país* del usuario. Los operadores usados para construir esta validación se muestran en la figura 2.

Para este algoritmo sólo se puede aplicar la medida de similitud del coseno. Las medidas de calidad fueron: RMSE = 1.159, MAE = 0.945 y NMAE = 0.236.

4.4. Algoritmo *Item Attribute k-NN*

Con este algoritmo se aplica la técnica de recomendación basada en características del ítem, en este caso se basa en el atributo *artista* de la canción. La validación de este algoritmo fue implementada de manera similar a la anterior, sólo que para este caso se usa el operador *Item Attribute k-NN*. Igual que en el caso anterior, en este algoritmo sólo se puede emplear la medida de similitud del coseno. Los resultados fueron: RMSE = 1.041, MAE = 0.839 y NMAE = 0.21.

5. Recomendación de Items

Este tipo de recomendación produce una lista de canciones.

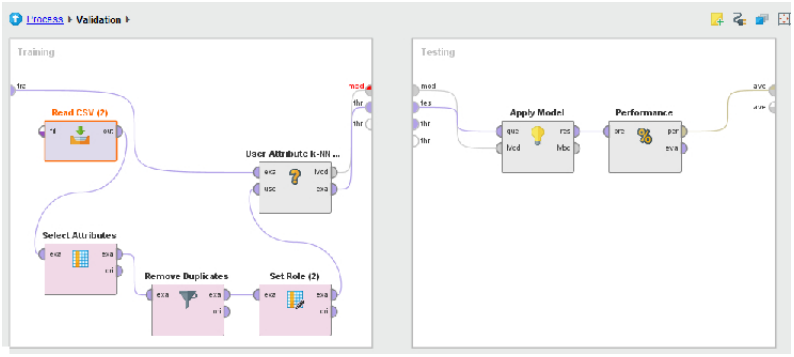


Figura 2: Recomendación basada en Rating: User Attribute k-NN

5.1. Algoritmo *User k-NN*

De la misma forma que el algoritmo *User k-NN* aplicado en la recomendación basada en *rating*, con este algoritmo se aplica la técnica de filtrado colaborativo de los vecinos más cercanos basado en el usuario. La validación fue implementada como se muestra en la figura 3.

Figura 3: Recomendación de Items: User k-NN

Al aplicar este algoritmo de recomendación, las medidas de calidad de las predicciones fueron: AUC = 0.754, NDGC = 0.238 y MAP = 0.083.

5.2. Algoritmo *Item k-NN*

Al igual que el algoritmo *Item k-NN* aplicado en la recomendación basada en *rating*, con este algoritmo se aplica la técnica de filtrado colaborativo de los vecinos más cercanos basado en ítems. La manera en cómo se implementó esta validación es similar a la anterior, sólo que para este caso se usa el operador *Item k-NN*. Se aplicó este algoritmo de recomendación y generó las siguientes medidas de calidad: AUC = 0.59, NDGC = 0.196 y MAP = 0.055.

5.3. Algoritmo *User Attribute k-NN*

Al igual que el algoritmo *User Attribute k-NN* aplicado en la recomendación basada en Rating, con este algoritmo se aplica la técnica de recomendación basado en el *país* del usuario. La validación fue implementada como se muestra en la figura 4.

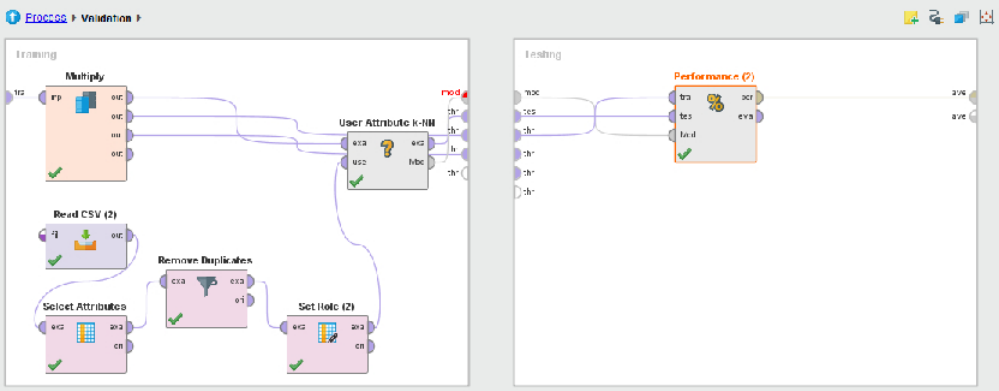


Figura 4: Recomendación de Items: User Attribute k-NN

Al aplicar este algoritmo de recomendación se generaron las siguientes medidas de calidad: AUC = 0.439, NDGC = 0.116 y MAP = 0.005.

5.4. Algoritmo *Item Attribute k-NN*

Al igual que el algoritmo *Item Attribute k-NN* aplicado en la recomendación basada en *rating*, con este algoritmo se aplica la técnica de recomendación basada en contenido, en este caso en el *artista* de la canción. La validación de este algoritmo fue implementada de manera similar a la anterior, sólo que para este caso se usa el operador *Item Attribute k-NN*. Al aplicar este algoritmo de recomendación, las medidas de calidad fueron: AUC = 0.773, NDGC = 0.256 y MAP = 0.091.

5.5. Algoritmo Híbrido: *User k-NN* + *Item Attribute k-NN*

Se han combinado los siguientes algoritmos aplicados de forma independiente anteriormente: *User k-NN* e *Item Attribute k-NN*. Los operadores usados para construir esta validación se muestran en la figura 5.

Se aplicó este algoritmo de recomendación con diferentes valores de k (30,40,50,80). Las medidas de calidad para cada una de ellas fueron: Con k=80 AUC = 0.837, NDGC = 0.333 y MAP = 0.145 - con k=50 AUC = 0.806, NDGC = 0.335 y MAP = 0.146 - con k=40 AUC = 0.805, NDGC = 0.337 y MAP = 0.148 - y con k=30 AUC = 0.793, NDGC = 0.337 y MAP = 0.149.

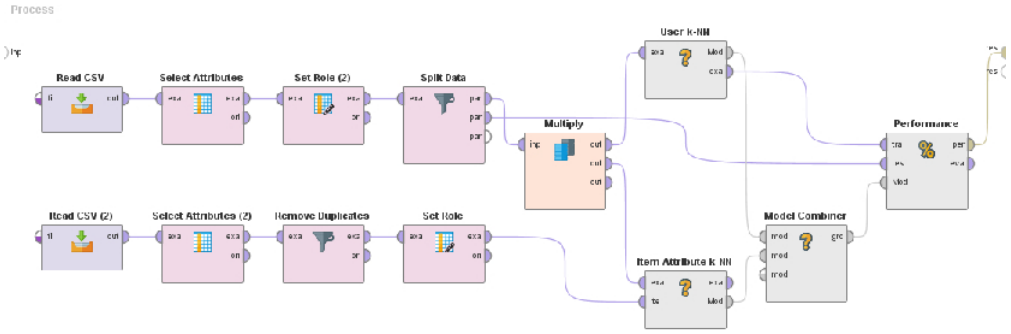


Figura 5: Recomendación de Items: User k-NN e Item Attribute k-NN

6. Evaluación de Resultados

La figura 6 muestra las medidas de error normalizado por cada algoritmo utilizado en la predicción de *ratings*.

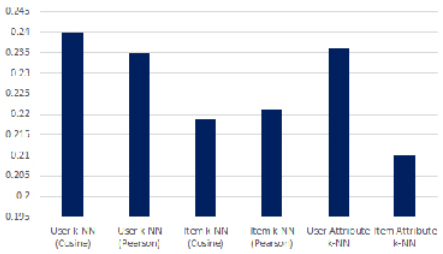
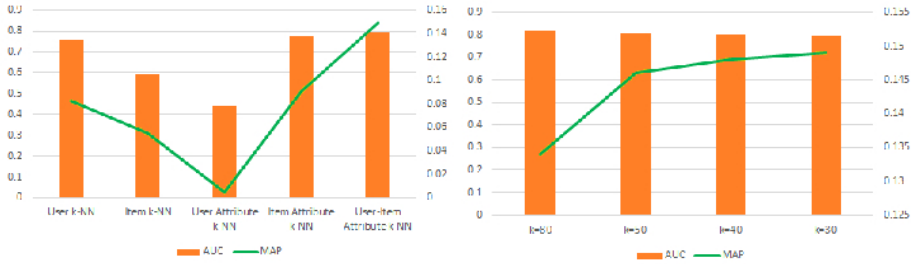


Figura 6: NMAE de cada algoritmo de recomendación basada en rating

Se puede observar que la medida de error es menor al aplicar el algoritmo *Item-Attribute k-NN* con la métrica de similitud coseno. Este algoritmo es un método basado en contenido puesto que hace uso de un atributo del ítem. Para este trabajo se usó el *artista* (atributo de la canción) para realizar la predicción del *rating*. Mientras que el cuadro 2 muestra las medidas de rendimiento AUC y MAP que se generaron al aplicar cada uno de los algoritmos para recomendar ítems.



Cada algoritmo con k=30 Modelo híbrido variando el k

Tabla 2: Medidas de calidad: AUC y MAP

En la gráfica izquierda se puede notar que el modelo híbrido, que combina el método de filtrado colaborativo de los vecinos más cercanos basado en usuario (User k-NN) y el método basado en contenido (Item Attribute k-NN con el atributo artista de la canción), proporcionó mayores valores de las medidas de calidad que el resto de técnicas aplicadas. Asimismo, los valores más bajos de estas medidas se obtienen cuando se aplica la técnica basada en el atributo del usuario (user Attribute k-NN con el atributo pais). En la gráfica derecha se puede apreciar que a medida que el valor de k disminuye, la medida MAP aumenta y AUC disminuye ligeramente, manteniendo un valor alto en torno al 80 %.

7. Conclusiones

En este trabajo se han aplicado diferentes técnicas de recomendación de filtrado colaborativo y de contenido, con el fin de generar listas de reproducción de música a partir de las recomendaciones obtenidas. El estudio experimental se ha realizado con datos obtenidos mediante la API de Last.fm. En primer lugar, se ha comprobado que en la categoría de los algoritmos de predicción de *rating*, el método de recomendación basado en contenido, que utiliza como atributo el artista de la canción, genera una mejor predicción que las demás técnicas. En segundo lugar, al aplicar los algoritmos de recomendación de ítems, se observó que el modelo híbrido, obtenido mediante la combinación de un método basado en usuario y otro basado en contenido, es el que clasifica mejor las piezas musicales, porque además de considerar el comportamiento del usuario, tiene en cuenta al artista de la canción. En base a estos dos enfoques se deduce que el artista es un aspecto importante del contenido de la canción. Por el contrario, el país es un atributo del usuario que no añade valor a la recomendación. Hubiera sido interesante contar con atributos del usuario como la edad o género para observar el comportamiento de los algoritmos basados en el atributo del usuario (User Attribute k-NN), sin embargo estas pruebas no se pudieron llevar a cabo porque estos atributos no estaban disponibles en el API de Last.fm.

Referencias

1. Nicholas J Belkin and W Bruce Croft. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992.
2. Dmitry Bogdanov, Martín Haro, Ferdinand Fuhrmann, Emilia Gómez, and Perfecto Herrera. Content-based music recommendation based on user preference examples. *Copyright Information*, page 33, 2010.
3. Ricardo Dias and Manuel J Fonseca. Improving music recommendation in session-based collaborative filtering by using temporal context. In *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*, pages 783–788. IEEE, 2013.
4. Marcos Aurélio Domingues, Fabien Gouyon, Alípio Mário Jorge, José Paulo Leal, João Vinagre, Luís Lemos, and Mohamed Sordo. Combining usage and content in an online recommendation system for music in the long tail. *International Journal of Multimedia Information Retrieval*, 2(1):3–13, 2013.
5. Thomas Hornung, Cai-Nicolas Ziegler, Simon Franz, Martin Przyjaciel-Zablocki, Alexander Schätzle, and Georg Lausen. Evaluating hybrid music recommender systems. In *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 01*, pages 57–64. IEEE Computer Society, 2013.
6. Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272. Ieee, 2008.
7. Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010.
8. Miquel Montaner Rigall et al. *Collaborative recommender agents based on case-based reasoning and trust*. Universitat de Girona, 2003.
9. Maciej Pacula. A matrix factorization algorithm for music recommendation using implicit user feedback. 2009.
10. Sung Eun Park, Sangkeun Lee, and Sang-goo Lee. Session-based collaborative filtering for predicting the next song. In *Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on*, pages 353–358. IEEE, 2011.
11. Waleed Reafee, Naomie Salim, and Atif Khan. The power of implicit social relation in rating prediction of social recommender systems. *PloS one*, 11(5):e0154848, 2016.
12. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.
13. J Ben Schafer, Joseph A Konstan, and John Riedl. E-commerce recommendation applications. In *Applications of Data Mining to Electronic Commerce*, pages 115–153. Springer, 2001.
14. Ja-Hwung Su, Wei-Yi Chang, and Vincent S Tseng. Personalized music recommendation by mining social media tags. *Procedia Computer Science*, 22:303–312, 2013.
15. Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems*, pages 2643–2651, 2013.

16. Zhe Xing, Xinxi Wang, and Ye Wang. Enhancing collaborative filtering music recommendation by balancing exploration and exploitation. In *ISMIR*, pages 445–450, 2014.

Visualización de datos en Humanidades Digitales

Alejandro Benito y Roberto Therón

Universidad de Salamanca
Departamento de Informática y Automática
Facultad de Ciencias
Plaza de los Caídos s/n
37008 Salamanca, Spain

abenito@usal.es, theron@usal.es,
http://vis.usal.es/~visusal/grupo/

Resumen El estudio de diccionarios históricos ha sido una tarea tradicionalmente compleja por la gran cantidad de fuentes y datos de diferente naturaleza que están implicados en el proceso. El análisis visual exploratorio puede ser de gran ayuda para las investigadoras en la materia a la hora de crear mapas mentales que reflejen acertadamente la situación de los datos, mejorando y acelerando la extracción de conocimiento de valor y la llegada a conclusiones significativas. En este artículo se proponen un método de trabajo y un prototipo para la exploración visual interactiva de corpus no estándar relacionado con los dialectos del bávaro en Austria. A partir de la importación de conjuntos masivos de datos provenientes de la digitalización de fuentes originales, logramos ofrecer a la investigadora en lexicografía una herramienta que logra dotar de un enfoque nuevo a los datos con respecto a otras soluciones existentes relacionadas y que sirve de marco de referencia para futuras colaboraciones en la materia entre equipos especializados en C. de la Computación y Lingüística.

Keywords: análitica visual, humanidades digitales, visualización de la información, lexicografía, dialectología

1. Introducción

Las Humanidades Digitales (en adelante HD) son un campo de estudio resultante de la intersección entre las disciplinas de las ciencias de la computación y las humanidades. En ellas se comprenden una serie de ramas o especializaciones, que varían desde la catalogación de colecciones online hasta por ejemplo, la minería de datos de grandes conjuntos de datos culturales de todo tipo. Las HD incorporan datos digitalizados y/o digitales y combinan metodologías de disciplinas tradicionales de humanidades (como la historia, filosofía, lingüística, literatura, arte, arqueología, música y estudios culturales) con herramientas proporcionadas por las ciencias de la computación (Ver Figura 1 izquierda).

En nuestro enfoque proponemos una solución visual al problema al estudio de los diccionarios históricos de dialectos del bávaro en Austria, mediante una

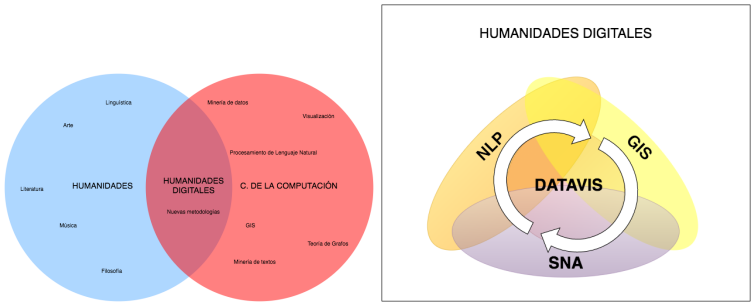


Figura 1: Izquierda: Esquema de la colaboración entre Humanidades y Ciencias de la Computación, resultante en la disciplina de las Humanidades Digitales. Derecha: Los 3 + 1 pilares computacionales de las Humanidades Digitales. En el centro, la Visualización de Datos sirve para conectar las otras 3 partes y exponer cada una de sus características a la usuaria final.

aproximación al problema asentada en pilares computacionales, todos ellos intercomunicados y expuestos a la usuaria final gracias a la aplicación de técnicas de visualización de la información (Figura 1 derecha). Estas ramas son: **1. GIS o Sistemas de Información Geográfica**: Tradicionalmente, el ser humano tiene una larga y rica historia relacionada con la representación de conceptos abstractos en mapas. Hoy en día, las representaciones en papel o pergamino han sido sustituidas por el uso de estos sistemas que ha venido de la mano con un incremento en calidad y cantidad de la información que puede ser codificada en los mismos. **2. NLP o Procesamiento del Lenguaje Natural**: En esta disciplina incluiremos técnicas de minería de textos y recuperación de la información, que tendrán el objetivo de acceder, condensar y abstraer el conocimiento subyacente en diferentes tipos de flujos de textos, normalmente a gran escala. El gran auge que han experimentado los motores de búsqueda textual en los últimos años ha motivado que estas tareas se hayan suavizado en dificultad considerablemente. **3. SNA o Análisis de Redes Sociales**: Las teorías sobre grafos, y los algoritmos de división, *pathfinding*, recorrido y balanceo van a ser particularmente útiles en esta temática. Y **4. Visualización de la Información**: La visualización es de vital importancia en el proceso debido a que va a proporcionar una entrada accesible y menos intimidatoria a la usuaria final a cualquiera de las otras tres ramas mencionadas anteriormente y por tanto, de la buena praxis del investigador al aplicar adecuadamente las técnicas de visualización va a depender que se libere todo el poder de las anteriores en la aplicación práctica de nuestra solución. Las visualizaciones elegidas en la implementación final de nuestro prototipo están en línea con las tendencias de la mayoría de investigadores en el área de las HD[12] y de otros trabajos relacionados en la materia, como se expone en la siguiente sección.

2. Trabajos relacionados

Los avances tecnológicos de los últimos tiempos, y el incremento en la accesibilidad a los mismos, han generado una gran producción académica y no académica en las HD en los últimos tiempos. Numerosas son las instituciones públicas y privadas que han puesto sus ojos en esta antigua disciplina renovada, tratando un gran número de temáticas diferentes, que varían desde el procesamiento y visualización de textos, hasta la representación de características, análisis de sentimientos y cultura tradicional en entornos GIS. La historia de las HD se origina en los años 40 gracias a los esfuerzos del sacerdote jesuita Roberto Busa, que comenzó la tarea de crear en soporte digital una transcripción de textos medievales de Sto. Tomás de Aquino[3]. En esta práctica de la transcripción digital de textos se basó gran parte del trabajo de expertas e investigadoras durante las dos décadas siguientes. De especial interés para este artículo son las producciones de Wilhem Ott y su grupo de investigación en la Universidad de Tübingen (Alemania) en los años 60, que resultaron en la creación de una *suite* de programas para el análisis de textos aún usada en nuestros días. Estos módulos, llamados TUSTEP[9], asentaron un formato orientado al manejo de textos y mucha de la información recopilada en la actualidad usada para el estudio de diccionarios en lengua alemana se encuentra codificada en este formato. En los años 90, este formato recibiría la influencia de lenguajes de marcado como SGML desarrollados al amparo de comités de expertos en HD y el procesamiento de textos (Text Encoding Initiative), adoptando también el estándar ISO XML para su representación.

Existen también en los últimos tiempo iniciativas para crear acercamientos visuales al problema del tratamiento visual de este tipo de información, como el trabajo de Thomas Mayer et al., titulado "An Interactive Visualization of CrossLinguistic Colexification Patters"[6]. Su enfoque trata de representar visualmente tendencias en el uso de ciertas palabras para referirse siempre a los mismos conceptos en los mismos idiomas, y ver su evolución en el tiempo, así como comparar dichas tendencias en un lenguaje con las de otros lenguajes. En la misma línea de destaca el artículo de Roberto Therón et al., "Diachronic-information visualization in historical dictionaries"[10], que también ahonda en el análisis visual de información diacrónica albergada en un corpus textual proveniente de diferentes diccionarios históricos del castellano. En esta investigación se propone una solución que permite al investigador visualizar la evolución de la posición de los diferentes significados de un mismo lema a lo largo del tiempo, empleando para ello una serie temporal animada y un mapa.

Destaca también la iniciativa del equipo de Daniel A. Keim de la Universidad de Constanza en Alemania para el análisis visual de patrones esta vez de series temporales de datos textuales. En el mismo se propone un *workflow* de dos pasos adaptado al entorno de aplicación del análisis de datos financieros [14]. Por último, el reciente trabajo de investigación propuesto por Roberto Therón y su grupo de investigación en colaboración con la Academia de las Ciencias de Austria[11], propone un análisis visual de corpus provenientes de diccionarios

históricos empleando proyecciones geográficas interactivas y sistemas de coordenadas paralelas.

3. Descripción del problema y conjunto de datos

El conjunto de datos empleado proviene de un proyecto de largo recorrido emprendido en 1911 por la Academia Austríaca de las Ciencias, el denominado Diccionario de los Dialectos Bávaros de Austria (WBÖ). En él se recopilan los léxicos empleados en diferentes partes del Imperio Austro-Húngaro y el Reino de Bavaria, con el objetivo de dar una visión detallada de los dialectos del idioma alemán, así como crear un precedente en el estudio de la lexicografía de los mismos. El método de trabajo tradicionalmente empleado por la Academia para registrar los usos del lenguaje se basaba en el uso de **cuestionarios**. En ellos se pedía a la población que respondiese a preguntas específicas sobre el léxico que empleaban para referirse a conceptos relacionados con cierta temática. La información recopilada por la persona encargada de esta tarea era también complementada con trabajo de campo, normalmente en forma de entrevistas personales. En ellas se concretaban los conceptos que hubiesen aparecido en los cuestionarios y que fuesen de especial interés lingüístico por diversas razones. Al final de este proceso la lexicógrafa trataba de dar una definición lo más exacta posible de cada término que iba a parar a una tarjeta. Estas tarjetas son también especialmente importantes por el hecho de que muchas veces (dependiendo de la habilidad de la lingüista), contenían dibujos descriptivos que ayudaban a contextualizar el concepto. Por ello eran recopiladas, ordenadas y almacenadas en registros junto a los cuestionarios y servían de base para futuras investigaciones. Estas tarjetas, junto a información suplementaria recopilada durante el proceso de investigación, eran transcritas empleando la suite de procesamiento de textos TUSTEP (Ver Figura 2).

Empleando estas herramientas de procesamiento de textos, las investigadoras realizaban consultas en las que basaban su extracción de conocimiento. Esta metodología por tanto se sustentaba únicamente en el empleo de métodos computacionales para la búsqueda de cadenas que eran combinados con métodos manuales en los que era muchas veces necesario acceder al registro físico en papel para generar artefactos complementarios que ayudasen en la investigación (mapas, esquemas y listas de palabras), lo cual hacía del proceso una ardua tarea a desempeñar. Estos datos, que vienen siendo progresivamente digitalizados y clasificados desde el año 2010[13] gracias a proyectos como el "Datenbank der bairischen Mundarten in Österreich" (DBÖ), suponen una valiosa fuente de información para historiadores y lexicógrafos en nuestros días, y es por tanto que se presenta el reto de ofrecer herramientas visuales que permitan el correcto acceso a los mismos. Esta información codificada en formato XML supone el primero de los subconjuntos de datos que empleamos en nuestro estudio. En el reciente proyecto dbo@ema (DBÖ Electronically Mapped) se lanza la iniciativa de poner a disposición de investigadores y público en general parte de estos datos en una plataforma web. Durante este proceso se realiza un proceso de *geocoding* en el

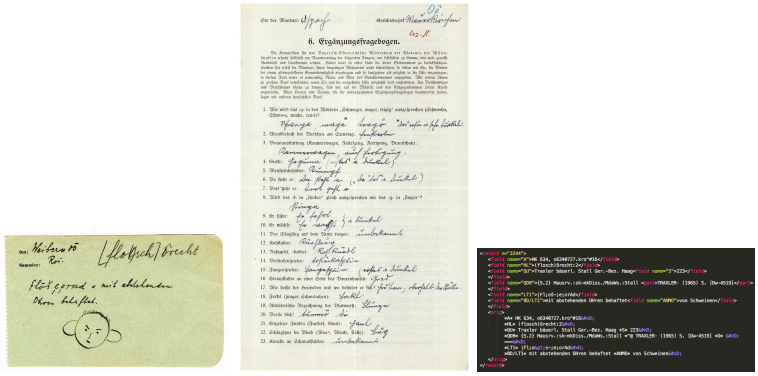


Figura 2: Izquierda: Una tarjeta con notas y un dibujo que contextualizan la palabra definida: “floschorecht” (orejudo). Izquierda: Copia escaneada de un cuestionario de principios de siglo realizado en la región de la Alta Austria. Derecha: Detalle del registro TUSTEP que hace referencia a la tarjeta. En el campo “BD/LT1” que hace referencia al significado, se lee la transcripción de la nota original: “mit wegstehenden Ohren behaftet” (afectado por orejas grandes).

que se dota a los datos de una componente espacial mediante el uso de técnicas GIS, que resulta en la creación de una BD que contiene esta información y que supone el segundo de los conjuntos de datos con los que se ha trabajado.

4. Desarrollo de la propuesta

Recordemos que en un principio se disponía de dos conjuntos de datos en diferentes formatos: MySQL (proyecto dbo@ema) y ficheros XML (fuentes originales TUSTEP). Ya que la naturaleza de los mismos hacía imposible su tratamiento y aplicación directa de técnicas de análisis visual se combinó información proveniente de ambos conjuntos de datos para crear un tercero que contuviese la información esperada. El objetivo pues fue conjugar e indexar toda la información disponible en una instancia del motor de búsqueda ElasticSearch, añadiendo las dimensiones textual, espacial y temporal cuando fuese posible, que permitiesen plantear un análisis visual multidimensional de la información.

En nuestro enfoque se emplea el conjunto de datos XML como fuente primaria de información, mientras que el conjunto proveniente de dbo@ema sirve de conjunto de datos de soporte al primero. El primer tipo de información que se deseaba incluir era la espacial ya que es en ésta que se centra el modelo de herramienta de análisis propuesto. Recordemos que el formato XML contenía también información espacial pero la misma no estaba geocodificada, ya que sólo se disponía del nombre del lugar al que hacía referencia la fuente. Por otro lado, la base de datos usada en dbo@ema sí que contenía tablas en las que se relaciona el nombre de un lugar tal como se puede encontrar en las fuentes XML de TUSTEP y su información GIS asociada. Por otro lado, la mayoría de la in-

formación se encontraba aún en formato XML. Era pues preciso combinar estas dos fuentes de información, extrayendo la valiosa información GIS de la base de datos MySQL e insertarla dinámicamente en los nuevos documentos creados en ElasticSearch junto con la proveniente de los registros TUSTEP. Veremos a continuación con un ejemplo este proceso de adquisición de dimensiones para un registro escogido al azar de entre los datos. En la Figura 3 se muestra una captura de pantalla ilustrando el contenido del registro XML y el registro de la BBDD que contendría la información espacial asociada.

1)
```
<record n="447">
   <field name="A">HK 869, z8690113.kro^#8</field>
   <field name="HL">(Hals)zapfen:1</field>
   <field name="QU">Blaindorf Stmk. Fabiani</field>
   <field name="QDB">{3.5g02} uFeistritzt.:mOSt. <part>FbB.FABIANI· (u.1913) [SFb.]</part>
      <field name="O">Blaindf. St.</field>
   </field>
   <field name="NR">4L7: Gaumen</field>
   <field name="LT1">H;olsz-abfrl [D2]</field>
   <orig>
      *A* HK 869, z8690113.kro^#8&#xD;
      *HL* (Hals)zapfen:1&#xD;
      *QU* Blaindorf Stmk. Fabiani&#xD;
      *QDB* {3.5g02} uFeistritzt.:mOSt. *^@ FbB.FABIANI· (u.1913) [SFb.] *O* Blaindf. St.&#xD;
      ===&#xD;
      *NR* 4L7: Gaumen&#xD;
      *LT1* H;olsz-abfrl [D2]&#xD;
      ===&#xD;
   </orig>
</record>
```

2)
```
id,nameKurz,nameLang,sort,bearbeitungsgebiet_id,gemeinde_id,gis_ort_id,namensvarErl,behoerde,quellen,ort_verzeichnis
_id,originaldaten,freigabe,checked,wordleiste,druck,online,publiziert,anmerkung,trust,menschkurz,OKZ,autokurz
16650,Blaindf.,Blaindorf,999999,2,995,15887,,,NULL,1,NULL,0,1,0,0,1,0,NULL,3,Blaindf.,15091,Blaindf.
```

Figura 3: 1: Registro 447 de un fichero XML de TUSTEP, referente al lema "Halszapfen". En el campo QDB podemos encontrar las dimensiones temporal y espacial asociadas a la fuente. 2: Representación CSV del registro asociado a Blaindorf en la BD MySQL. Obsérvese que la coincidencia de los nombres no es exacta.

En la figura 3 podemos observar los diferentes campos que componen este registro. De especial interés es el campo "QDB", que como vemos tiene información temporal de una fuente (en este caso un cuestionario, FbB) e información espacial contenida en un subcampo "O" emplazado a tal efecto. Vemos también que no existe una correlación directa entre el nombre del topónimo en XML y su representación en MySQL, haciendo imposible la automatización completa del proceso.

4.1. Extracción de la dimensión espacial

Mediante la aplicación de técnicas heurísticas adquiridas mediante el estudio de los datos, se llegó a la conclusión de que la mejor opción era implementar una

serie de reglas que codificasen el conocimiento experto adquirido. Cuando estas reglas no fuesen capaces de asociar correctamente las dos entradas provenientes de ambos conjuntos de datos, se requeriría de la intervención humana para finalizar el proceso, bien creando la asociación manualmente o bien descartando cualquier asociación, quedando el registro final sin información GIS asociada.

En un principio se contabilizan 1.861.878 registros (80 %) con algún tipo de referencia espacial asociada no estandarizada (la referencia al lugar no se encuentra dentro del registro "QDB". 322.459 registros (14 %) contienen dicho campo. Por otro lado, no siempre la información espacial hace referencia a un punto en el mapa: En otros casos se hace referencia a las divisiones administrativas de Austria: una comunidad o una región, dando lugar a diferentes resoluciones espaciales.

4.2. Extracción de la dimensión temporal

En el caso de la información temporal, se codificaron también reglas heurísticas para extraer dicha dimensión. Esta información puede venir en forma discreta o de intervalo cuando la fuente es por ejemplo un volumen que se extiende a lo largo de un período de tiempo, o cuando la fecha de origen exacta es desconocida y se proporciona una estimación. Para resumir, la información temporal va a venir representada en varios formatos, cada uno con una interpretación asociada diferente. Un 71 % de los registros presentan información temporal asociada.

- 1945 (Cuatro dígitos, rodeados por un número variable de caracteres): Ofrecen el máximo nivel de resolución temporal, el año. Se recuperan 509.929 registros, que suponen un 22 % del total.
- 1945-50 (Cuatro dígitos + guión + dos dígitos): Resolución temporal menos de un año pero mayor que década. Se extraen 155.240 (6 %) registros.
- 193x (3 dígitos + "x"): La resolución temporal es de década. Suponen un 2 % (54.374) de todos los registros.
- 19xx (Dos dígitos + "xx"): La resolución temporal es de siglo. En los registros sólo se encontraron ocurrencias de este tipo que hacen referencia al siglo XX. Un 41 % (954.126) de los registros presentaban este formato.

4.3. Conjunto final

Al terminar el proceso de extracción de características que combina los dos conjuntos de datos, se indexada la información recabada en una instancia del motor de búsquedas para su posterior tratamiento visual. Es conjunto resultante arroja las siguientes cifras: 1: Se importan correctamente 2.206.227 registros (95.3 %) de los 2.314.031 originales. El resto son descartados por errores en el formato original de los datos. 2: De los registros importados correctamente, un 9.8 % contienen información referente a las dimensiones **espacial y temporal**. 3: Un 26.6 % contiene información **temporal** pero **no espacial**. 4: Un 32.4 % contiene información **espacial** pero **no temporal**. 5: un 31.1 % de los datos **no contiene** ningún tipo de dimensión **temporal** o **espacial**.

4.4. Prototipo visual propuesto

El prototipo resultante de la investigación es una herramienta de análisis visual multidimensional de la información recuperada y adaptada en la fase anterior de adquisición de datos. A pesar de que el enfoque es multidimensional, éste le da una mayor importancia inicialmente a la dimensión espacial, que en un análisis exploratorio en el que no dispongamos de ningún tipo de entrada por parte de la usuaria, va a ser la que sirva de guía al proceso. En nuestro enfoque, consideramos que el porcentaje de los datos generados en la anterior etapa de adquisición de los mismos es válido para guiar el flujo de trabajo por esta dimensión. En la Figura 4 recogemos una captura de la interfaz de la aplicación con todas sus vistas desplegadas:

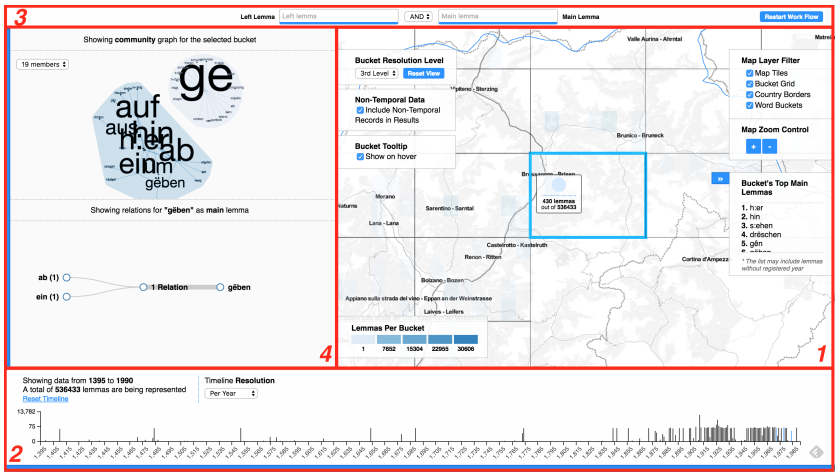


Figura 4: Interfaz del prototipo propuesto con 1) Proyección espacial o mapa, 2) Proyección temporal o *timeline*, 3) Barra de búsqueda textual, 4) Vista de análisis de redes

La interfaz muestra 4 vistas principales, de las que sólo 3 estarán disponibles en un principio (la vista 4 se muestra y oculta dinámicamente dependiendo de la fase del flujo de trabajo en la que se encuentre la analista en un momento dado). Como introdujimos en las secciones previas, este prototipo propone un flujo de trabajo basado en el mantra de la visualización, recordemos: "Visión general primero, zoom y filtrado, por último detalles en demanda". A continuación, basándonos en este precepto, describiremos el funcionamiento general del prototipo y cómo se aplican las técnicas de visualización y UX a cada una de las partes del flujo de trabajo para conseguir un análisis visual del diccionario.**Visión general primero:** La aplicación realiza su primera carga mostrando una vista general de los datos que sirve como punto de partida para un análisis visual exploratorio multidimensional y dirigido espacialmente.

Las técnicas de visualización en las que se realiza una primera carga de datos de todo el conjunto disponible suelen ser suficiente cuando el volumen de los datos no es excesivamente grande, y por tanto su tiempo de interacción es aceptable. En nuestro caso no es posible, sin embargo, aplicar estas técnicas sin comprometer la experiencia de usuario. Encontramos, por tanto, los siguientes problemas: 1. Se necesitan estructuras de datos que contengan métricas generales sobre los datos analizados en base a agrupaciones. 2. Estas métricas no pueden ser construidas en un proceso previo ya que se perderían muchas de las opciones de análisis que se podrían ofrecer sobre el conjunto de datos. Para ejemplificar este hecho, imaginemos la situación siguiente: Se quiere realizar un *bubble map* que agregue diferentes localizaciones geográficas de elementos en el mapa. En el momento que cambiemos los criterios de búsqueda o filtrado (empleando las vistas 2 o 3 en nuestro caso) necesitaremos recalcular todas estas métricas de nuevo en base a un nuevo subconjunto de resultados que encaje con dichos criterios. Esto lleva a al problema 3. Este tipo de operación no es factible en nuestro enfoque porque llevaría a a) Disponer de todos los datos en la primera carga y aplicar las métricas en tiempo real, lo cual es inviable con el volumen de información o b) Precalcular todas las métricas para todas las situaciones de filtrado y búsqueda posibles, lo que en nuestro ejemplo tampoco es viable, ya que este número de posibilidades es, a efectos prácticos, infinito. La solución a este problema la va a dar la instancia del motor de búsquedas, que en todo momento va a proporcionar dos estructuras de datos que van a ser la base de la visualización y que contienen datos a la resolución necesaria, maximizando así el rendimiento y el aprovechamiento de los recursos disponibles.

El **mapa** es la vista central sobre la que se basa el flujo de trabajo propuesto para el análisis exploratorio de los datos. Presentamos un detalle de la vista de mapa y de sus elementos en la Figura 5, de la que explicaremos a continuación sus diferentes partes:

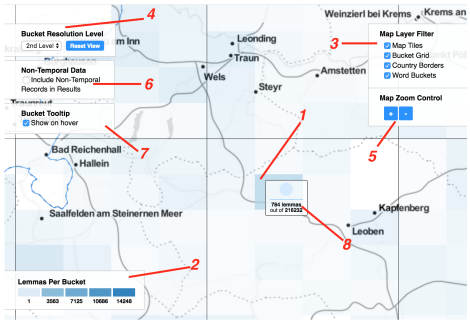


Figura 5: Detalle de la vista del mapa. 1)Geohash/Bucket espacial, 2)Escala, 3)Control de capas, 4)Control de resolución de los datos, 5)Control de zoom, 6)Control para incluir resultados sin información temporal, 7)Mostrar/Ocultar vista resumen del bucket, 8)Vista resumen del bucket.

Para la representación de las agregaciones empleamos la técnica de *geohashing*[7], un sistema de geocodificación que soporta búsquedas textuales. En este sistema, una cadena codifica una porción rectangular de terreno, de manera que cuanto más larga es la cadena, más pequeña es la porción que define, y por tanto más resolución se obtiene. Cada *geohash*, va a representar una agregación de todos los elementos que "caen" dentro de sus límites mediante una escala de colores que asocia el número de ocurrencias encontradas a tonalidades azules en orden creciente de oscuridad. (Figura 5.2). El mapa muestra diferentes capas, cada una aportando un tipo de información diferente. En nuestro prototipo vamos a emplear 4 capas diferentes: 1.Capa de *tiles*, que muestra las imágenes del mapa. En ellas se muestran las distintas localidades, orografía, etc. Esta capa es fundamental y sirve para contextualizar la información mostrada en las otras capas. 2.La capa de *grid* muestra el nivel anterior de resolución al elegido. Sirve también para contextualizar en este caso los *buckets* que se muestran en cada momento, aportando una referencia visual sobre cuál es el *geohash* "padre" de cada uno. 3.En la capa de fronteras se representan las fronteras de los diferentes países mostrados en el mapa. Esto es útil en el aspecto geográfico e histórico, ya que la analista tiene una referencia en todo momento de la posición de las fronteras actuales y así puede determinar si estas han variado o no desde el momento en el que se data una fuente, por ejemplo. 4.Por último la capa de *buckets* muestra la información de los términos en *buckets*, como se explicó anteriormente.

Todas estas capas pueden ser ocultadas o mostradas a petición de la usuaria. Esta ocultación permite que no sea distraída o molestada por elementos que no son relevantes en el momento de la exploración en el que se encuentre. (Figura 5.3). En las Figuras 6 y 7 podemos observar la misma porción del mapa mostrando datos a distinta resolución. En la marca número 4 de la Figura 5 el control que maneja la resolución en la vista del mapa. Al igual que se dispone de control de la resolución de los datos, se ofrece la opción de controlar el nivel de zoom, con resultados visuales semejantes.

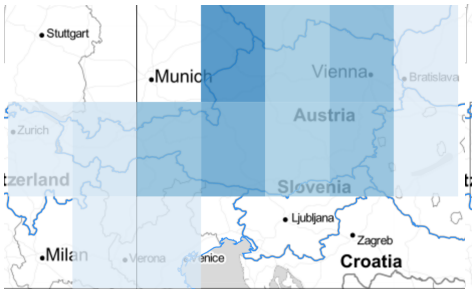


Figura 6: Menor resolución posible.

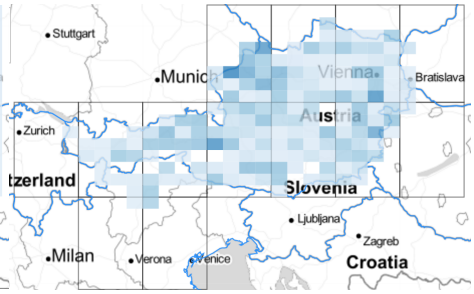


Figura 7: Un nivel más de resolución.

Ya que existen diferentes subconjuntos de los datos, cada uno con una o dos dimensiones asociadas, se decidió por defecto trabajar con el subconjunto

que contiene las dimensiones espacial y temporal asociadas en el mapa, y con el subconjunto que tiene información temporal (aunque no disponga de espacial) en el *timeline*. Sin embargo, no se podía ignorar el hecho de que los conjuntos que no contienen alguna de estas dimensiones pueden contener información valiosa para los investigadores y por tanto se decidió también dar la opción de incluir este subconjunto en los resultados por medio del control de la Figura 5.7. Por último cuando la usuaria desliza el puntero encima de un bucket, se muestra una vista resumen del mismo, que indica el número exacto de elementos recogidos en éste, así como el porcentaje que representa dentro del total de los resultados (Figura 5.8).

De manera análoga a la vista espacial del mapa se implementa la funcionalidad de la **línea temporal**. En ella se proyectan todos los datos que cuentan con este tipo de información. Se presenta un detalle de la misma en la Figura 8.

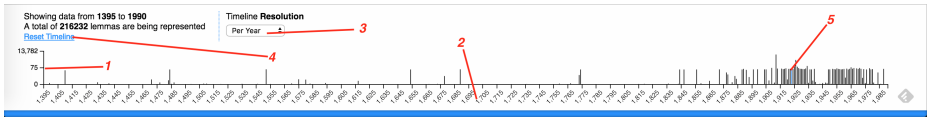


Figura 8: Detalle de la línea temporal. 1)Escala, 2)Representación de la dimensión temporal, 3)Control de resolución, 4)Texto explicativo y función de reset, 5)Barras y highlighting

En el eje X se aplica la dimensión temporal, mientras que en el Y se muestran el número de ocurrencias en cada año por medio de barras en base a la escala mostrada en la marca 1 de la figura. Esta escala relaciona la longitud de las barras con este número de ocurrencias (a más ocurrencias, la longitud de la barra es mayor, en base a un mínimo y a un máximo globales) y varía en función del subconjunto de datos con el que se trabaje. En la Figura 9 vemos la misma barra con sus escalas modificadas.

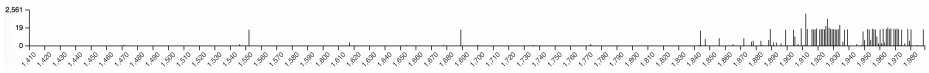


Figura 9: La línea temporal mostrando un nuevo conjunto de resultados. Nótese cómo varía la escala mostrada en el eje Y en base al mínimo y máximo encontrados, manteniendo la misma longitud. De manera análoga, el eje X muestra un nuevo conjunto de años en base a los mismos criterios.

En nuestro prototipo permitimos elegir entre 4 opciones de resolución de la línea temporal a través del control de la Figura 8.3. En la Figura a continuación mostramos una comparativa del mismo conjunto de datos mostrado a resoluciones de 1 y 25 años, respectivamente:

Figura 10: Conjunto de datos proyectado a resoluciones de 25 años (izquierda) y 1 año (derecha)

Por último se aporta una vista que muestra cifras que resumen lo que se está viendo en la línea temporal y el mapa: La cantidad de registros y los años mínimo y máximo encontrados en los mismos. También se incluye un control de reset, que elimina los filtros que se hayan aplicado a la línea temporal. La vista temporal soporta también la técnica de visualización denominada *highlighting*. En esta práctica, se resaltan en colores diferentes aquellas porciones de datos que se seleccionan en otras vistas enlazadas, como el mapa o el grafo en un intento de aligerar la carga cognitiva asociada al proceso de exploración.

En una segunda parte del flujo de trabajo se requiere la interacción por parte de la analista, que interaccionará con la aplicación a través de las observaciones realizadas en la primera gracias a las ayudas visuales explicadas. Es ahora donde la usuaria aplicará el filtrado y el zoom para reflejar su estado mental, centrado en una parte de los resultados mostrados. Nuestro prototipo soporta **tres tipos de filtrado** de los diferentes documentos: **1. Espacial**, a través de los elementos interactivos mostrados en el mapa; **2. Temporal**, empleando los recursos ofrecidos por el *timeline*; y **3. Textual**, que permitirá hacer búsquedas complejas de cadenas en el campo "HL" del registro original. El flujo de trabajo propuesto va a ir combinando estos tipos de filtrado en sucesivas etapas de refinamiento de los datos hasta que se produzca el descubrimiento de conocimiento. El **filtrado espacial** se consigue mediante la interacción con los *buckets* mostrados en el mapa. Cuando la usuaria, bien a través de la vista general ofrecida de los datos o bien con la ayuda de las vistas auxiliares que aparecen al deslizar el ratón sobre cada bucket, selecciona un bucket, hace explícito su interés por continuar su análisis en esa parte del mapa. Cuando este sucede, se generan una serie de acciones y animaciones en la interfaz que listamos a continuación: 1: Se realiza una acción de zoom sobre la zona geográfica comprendida por el bucket seleccionado. 2: Se cambia automáticamente la resolución del mapa a un nivel adecuado para dicho nivel de zoom. 3: Se recuperan elementos desde el motor de búsqueda y se muestran en los diferentes buckets. Se actualiza también la escala de colores del mapa. 4: Se presentan los elementos de análisis visual de redes en pantalla. 5: La interfaz refleja un nuevo estado mental de la usuaria. Es ahora donde se ofrece una nueva vista general de manera análoga a lo visto anteriormente, pero que emplea un subconjunto de los datos creado a partir de los de la anterior etapa. 6: El flujo se repite indefinidamente, quizás con la inclusión de otros tipos de filtrado o zoom provenientes de otros elementos de la interfaz. En la Figura 11 ilustramos este proceso. En un primer instante, una cierta zona del mapa llama la atención a la analista, que interacciona con el bucket (Arriba). Al pulsar sobre él, se lanza la cadena de eventos que termina con el estado de la interfaz mostrado a la derecha.

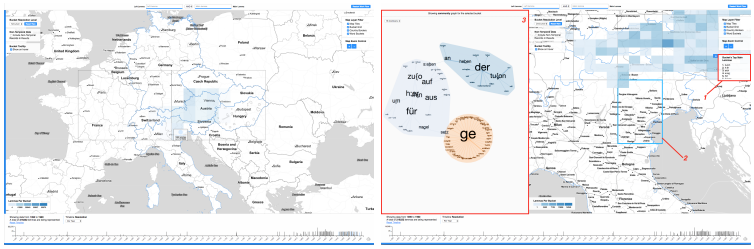


Figura 11: Los dos estados de la interfaz antes y después de realizar el filtrado espacial.

En las marcas 1 y 3 se muestran los nuevos elementos de la interfaz correspondientes al análisis de redes. En 1, los lemas (nodos) más importantes, en base al número de relaciones con otros lemas, son mostrados a la usuaria. En 3, una representación visual de las relaciones encontradas en forma de grafo dirigido de fuerzas. En 2 se añade una ayuda visual que permite a la usuaria recordar qué bucket de nivel inmediatamente superior fue pulsado al inicio de la interacción.

Además del filtrado espacial, la usuaria puede decidir trabajar con un subconjunto de los datos elegido a través de la realización de un **filtrado por la variable temporal**. Al existir una relación entre las dimensiones en memoria, este tipo de filtrado no necesita de nuevas peticiones y es, por tanto, sensiblemente más rápido. Este filtrado se consigue mediante la acción de arrastrar, como es común en este tipo de elemento visual. La zona elegida se representa por un sombreado que, al modificarse actualiza los datos mostrados en el mapa. Una vez fijado el filtro temporal, se mantiene en etapas subsecuentes de refinamiento si no es modificado por la usuaria. En la Figura 12 se muestra el mismo conjunto de datos filtrado primeramente en el intervalo formado por los años 1404-1932 (izquierda) y 1932-1987 (derecha). Obsérvese cómo la distribución espacial cambia en el mapa, así como la escala de colores, para reflejar cada uno de los dos subconjuntos formados.

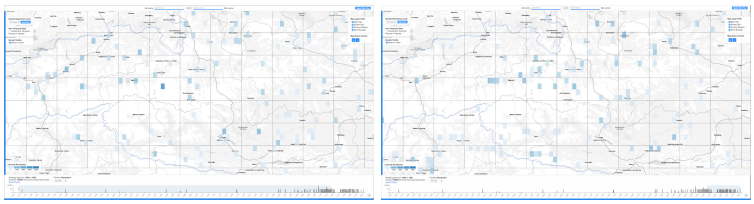


Figura 12: Dos capturas de la interfaz mostrando filtrados temporales en intervalos diferentes para el mismo conjunto de datos.

Por último el **filtrado textual** en tiempo real supone uno de los avances más importantes de la investigación. Gracias a esta capacidad, la analista puede buscar patrones de coincidencia entre fuentes coincidentes con los criterios de búsqueda textual introducidos. También debido al modelo elegido, la interfaz se mantiene en un estado responsivo en todo momento y permite una interacción dinámica y natural con los datos que reduce en gran medida el esfuerzo necesario inherente a la tarea de análisis. La sintaxis de búsqueda de *Lucene* basada en expresiones regulares, implementada por el motor de búsqueda va a soportar un extenso conjunto de operaciones lógicas, difusas o de proximidad que son de alto valor en el ámbito del estudio de la lexicografía. En una primera versión del prototipo, soportamos la búsqueda por las diferentes partes del lema, que recordemos habíamos denominado *leftLemma* y *rightLemma*. Para combinar la búsqueda por estos dos campos, se habilita también un selector lógico booleano AND/OR, que permite combinar ambas de dos formas diferentes. En cada una de las cajas de búsqueda empleamos una característica de la sintaxis *Lucene* y las combinamos mediante operadores lógicos diferentes para lograr conjuntos de resultados diferentes. La combinación de los dos criterios escogidos genera diferentes representaciones visuales de los conjuntos de datos. Ésto hace que sea sencillo para la analista acceder a una funcionalidad compleja del motor búsqueda, así como crear un mapa mental de la situación planteada por la inclusión de parámetros lógicos.

Ya se apuntó en los apartados anteriores la posibilidad de crear estructuras visuales para el **análisis de redes**. Dichas estructuras son generadas a partir de los conjuntos de datos que encajen con los parámetros de búsqueda introducidos por la analista, bien automáticamente o bien a petición de la usuaria, dependiendo del caso. Estas estructuras suponen el final de una iteración en el flujo de trabajo de la analista, ya que son capaces de desvelar relaciones ocultas entre los lemas imposibles de descubrir por medio de otros tipos de análisis (espacial o temporal). A partir de elementos interactivos, la analista va a poder refinar su flujo de trabajo y adaptar los datos a las coincidencias encontradas mediante la exploración visual de las redes de lemas. El primer tipo de representación visual que se presenta a la analista para el análisis visual de redes es el **grafo dirigido de fuerzas**, que es combinado con el enfoque de nube de palabras, para transmitir la idea de dos valiosos conceptos: la **importancia** de un lema y la **pertenencia** a un grupo del mismo. Esta visualización presenta varias características: 1: Genera un grafo de los elementos que entran dentro del criterio de búsqueda de la analista (una combinación de filtrado textual, espacial y textual). 2: En él se representan las relaciones entre lemas, en base a sus partes izquierda y derecha. Un lema que aparece en una entrada a la izquierda de otro aparecerá en el grafo como origen de la arista que los une. 3: Cada lema está representado en el grafo por las letras que lo componen, que formarán el nodo. Este nodo variará en tamaño en base a un escala lineal que relaciona el tamaño con la **importancia** del lema en el conjunto o lo que es lo mismo, en base al número de relaciones (aristas que llegan o parten) de ese nodo. 4: Debido al gran tamaño que pueden presentar estos grafos, especialmente en la búsqueda exploratoria, se genera además un análisis de comunidades sobre dicho grafo,

método mencionado en el trabajo de Mayer et al.(2014)[6] para la búsqueda de patrones de colexificación. 5: Se aplica además un filtrado dinámico en base al tamaño medio de las comunidades detectadas en el grafo, que permite dar una primera visión del grafo lo más adecuada posible a la analista. Nuestro flujo propuesto encuentra comunidades no superpuestas de nodos en los grafos con el objetivo de realizar particiones de los mismos que aporten valor a la investigación a la hora de la búsqueda de patrones reconocibles en los datos. En este tipo de particiones del grafo, la red se divide de forma natural en grupos de nodos densamente conectados internamente y con pocas conexiones con otros grupos. El algoritmo de detección de comunidades empleado es el denominado *Louvain*[2], suficientemente acertado y veloz[8] para poder ejecutarse en nuestro entorno. En la Figura 13 mostramos el grafo generado para la región delimitada por el geohash "u20", donde podemos ver las distintas comunidades representadas visualmente mediante la técnica de *Convex Hull*, cada una con un color proveniente de una escala categórica creada a tal efecto.

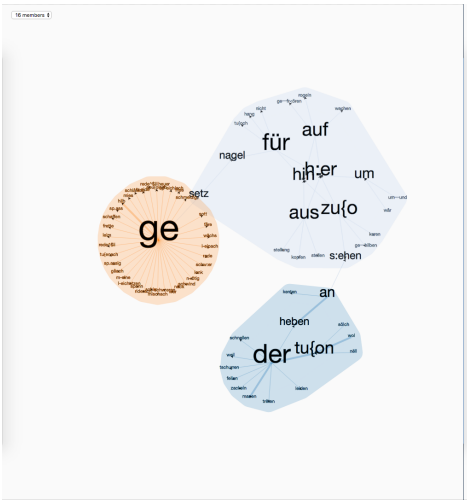


Figura 13: Vista inicial del grafo, con el filtro activado por defecto a 16 miembros.

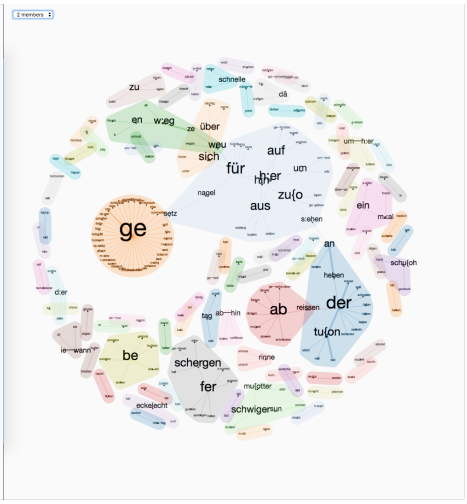


Figura 14: El grafo con el nivel de filtro activado a 2 miembros, muestra comunidades menos relevantes

Como se aprecia en las dos imágenes adjuntas, el grafo es filtrado mediante el control de la esquina superior izquierda. Este control se activa por defecto al valor medio del tamaño de las comunidades encontradas, de manera que las comunidades menos importantes son ocultadas al inicio. En la Figura 14 se ha modificado este valor a petición de la analista, mostrando el resto de elementos del grafo. Las comunidades más grandes, ocupando más espacio, desplazan a las menos importantes hacia los extremos del lienzo sin comprometer la expresividad de la visualización. Este tipo de representación permite también interactuar

con los nodos, así como hacer zoom y desplazamiento, como en el caso del mapa. En esta vista la analista va a poder analizar de manera visual las diferentes comunidades creadas y sus relaciones, creando un mapa mental del conjunto analizado que va a ayudar a la investigación y por tanto a la llegada a conclusiones significativas.

En la última parte del flujo de trabajo propuesta es donde la usuaria, a través de las vistas globales identifica un hecho significativo como cierta predominancia de una clase de lemas a originarse en partes concretas del mapa, ciertas relaciones entre lemas que tienden a repetirse, etc. La usuaria, inconscientemente, ha fijado su estado mental en este hecho y es por tanto necesario proporcionarle la opción de actualizar la interfaz en concordancia con el mismo. En nuestro enfoque existen diferentes maneras específicas de lograr esto (además de modificar alguna de las ya citadas más generales), con la peculiaridad de que todas ellas necesitan de la interacción de la usuaria (a diferencia de las anteriores, que sucedían de un modo más o menos automático).

Aparte del análisis de comunidades, que se aplica a grupos de lemas, se añadió la opción de realizar otro análisis visual orientado a un sólo lema, que también representase la red formada por el mismo en relación a otros. Recordemos cómo en la Figura 11.1 se mostraba, al seleccionar un bucket, una vista de detalle que incluía en orden decreciente de importancia los lemas encontrados en el mismo. Se ofrece a través del grafo de comunidades, la opción de generar el gráfico de árbol que permite navegar relaciones del lema elegido, bien en todo el conjunto de datos, o bien en la porción de datos seleccionada. El número de coincidencias con otros lemas se muestra junto a cada una de las ramas del árbol en el que el lema escogido es raíz. Esto permite una rápida exploración de las relaciones de un lema, que a su vez va a servir para generar nuevas búsquedas que refinen los resultados de la investigación. En la Figura 15 vemos el diagrama de árbol para el lema *milch* (leche):

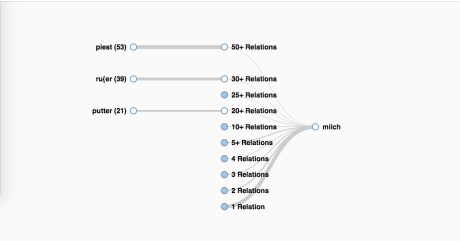


Figura 15: El gráfico de árbol que visualiza la red para el lema *milch* en la parte derecha o principal.

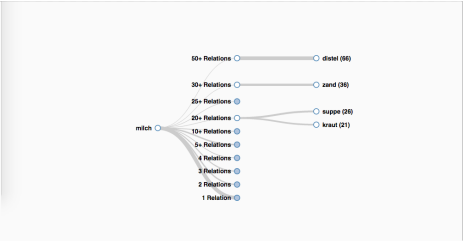


Figura 16: El gráfico de árbol con la red formada con *milch* en la parte izquierda del lema.

Una característica añadida a este análisis individual de lemas, es que el gráfico de árbol permite el intercambio de posición del lema escogido, mostrando coincidencias de resultados en los que éste aparece en la parte izquierda o de-

recha, a petición de la analista. En la Figura 15, se expande la red para relaciones con 20, 30 y 50 resultados. Para completar el proceso de análisis, se da la posibilidad de que la analista lance nuevas búsquedas interaccionando con los elementos del árbol, una vez que éste ha sido desplegado hasta un nodo hoja. En el caso presentado, al seleccionar una rama, se lanzaría una nueva búsqueda textual con *milch* como lema principal o izquierdo, según corresponda, y el otro término como parámetro complementario, comenzando así una nueva búsqueda. Simplemente con desplazar el cursor sobre las diferentes ramas, se emplea también *highlighting* en la proyección temporal, dando una idea del reparto de la relación en el tiempo. En el grafo de comunidades se permite, además de la generación de este grafo de árbol, lanzar nuevas búsquedas textuales en base a las comunidades detectadas o a nodos individuales del mismo que darán lugar a nuevas iteraciones de refinamiento del ciclo de trabajo.

En ocasiones, también puede resultar interesante ver el contenido original del registro TUSTEP con el objeto de complementar la información de la que dispone para un registro en concreto. Este tipo de información se muestra cuando la búsqueda ha reducido tanto el conjunto de datos que existen *buckets* con una sola instancia. Es entonces que al seleccionarlos se muestran los campos a la usuaria, como se recoge en la Figura 17.

Figura 17: Vista detalle que muestra los campos originales TUSTEP de un elemento de la visualización.

5. Caso de estudio: Detección de la homonimia

Se muestra en esta sección una aplicación práctica del prototipo en la búsqueda de patrones en la formación de nuevas palabras en los diferentes dialectos. Un fenómeno lingüístico que se da con cierta frecuencia y suele ser objeto de estudio en lexicografía dialectal es el de la degeneración fonética de los lemas. Dependiendo del acento que se emplee en un zona de influencia de cierta lengua, una palabra puede ser pronunciada de forma diferente a la original. Con

el paso de los años, esta nueva pronunciación puede adquirir tal importancia que el término original es reemplazado y el concepto al que éste se refería es verbalizado mediante una nueva palabra con entidad propia resultante de esta variación fonética. Esta nueva palabra reflejará en su escritura esta nueva pronunciación, que pasará a formar parte del léxico del dialecto hablado en dicha zona. Aplicando métodos computacionales[5] y visuales, se ayuda a la investigadora en la detección de patrones típicos de estos fenómenos y acelerarán por tanto la extracción de conocimiento de los datos.

5.1. Visión general primero

Estos patrones típicos suelen denotarse por la aparición de términos en los que los lemas original y degenerado aparecen en la misma posición junto a los mismos lemas, generando palabras parecidas en escritura pero con significados iguales. La analista va a comenzar la sesión de estudio mediante una búsqueda dirigida sobre el color rojo, cuyos motivos explicamos brevemente antes de pasar al detalle del proceso: Cuando se producen diferencias en la pronunciación de los términos, las palabras sufren variaciones en la escritura para representar este cambio. Sin embargo, como estas nuevas palabras derivan de una diferencia fonética, su escritura no resultará ser muy diferente y por lo tanto, la analista puede lanzar una **búsqueda difusa** en sintaxis *Lucene* que emplee la distancia de Levenshtein para encontrar palabras "parecidas" a la dada. En respuesta a esta petición del usuario, el motor de búsquedas va a devolver ocurrencias de lemas que tengan una distancia de Levenshtein de 1 (que necesiten de una edición para ser iguales a la proporcionada) con respecto a la original. En el caso del lema sometido a estudio "rôt", la analista sabe también, que éste aparece con mucha más regularidad en la parte izquierda que en la derecha, así que comenzará lanzando la búsqueda con esta premisa. En las Figuras 18 y 19 presentamos dos detalles del mapa de la interfaz, el primero capturado al introducir la palabra "rôt" y el segundo instantes después de añadir el caracter "~" al término (y eliminando el carácter fonético de la letra "o"), expresando su intención de realizar una búsqueda difusa. Además, gracias a la técnica de UX de búsqueda instantánea, la usuaria del sistema va a poder ir observando estos cambios de distribución espacial a la vez que teclea.

Al ampliar la búsqueda inicial del término para incluir la componente difusa de distancia entre palabras, se pueden apreciar varios hechos importantes en la visualización espacial; 1: Se amplia el número de resultados, que pasa de 638 a 1099. 2: Aunque no se muestre en las imágenes adjuntas, la distribución temporal cubre un período más amplio (aproximadamente 300 años mayor). 3: Existe una zona al norte del país que ha variado considerablemente su tonalidad al incluir el parámetro difuso en la búsqueda. Esto denota que esa zona ha sido especialmente afectada por la introducción de este cambio.

5.2. Zoom y filtrado

La analista, a vista de estos resultados y especialmente de la última conclusión de las apuntadas, va a aumentar la resolución de los *buckets* manualmente

Figura 18: Detalle de la distribución espacial de la búsqueda de “rôt”

Figura 19: Distribución espacial de la búsqueda difusa de “rot”

para poder observar en más detalle la distribución espacial de los resultados (Figura 20).

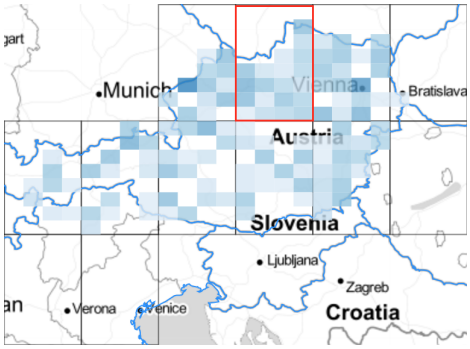


Figura 20: Detalle del mapa mostrando la distribución espacial de los resultados a resolución más alta.

La analista analiza nuevamente la situación y observa la distribución de los datos dentro del bucket seleccionado. A pesar de no ser la zona que cuenta con más ocurrencias, a este nivel de detalle sí permite afirmar que la distribución espacial de las mismas es la más amplia de todas. Esto, ligado al hecho de ser la zona más afectada por la introducción del parámetro difuso hace que la usuaria se decante por centrar su análisis en ella.

Cuando la animación termina, se ha generado un nuevo grafo que va a permitir realizar un análisis de redes sobre el subconjunto de los datos seleccionado que se muestra en detalle en la Figura 21.

Lo primero que llama la atención del grafo recién generado son las dos grandes comunidades dominadas por los términos “rôt” y “rotz” mucho más resaltados

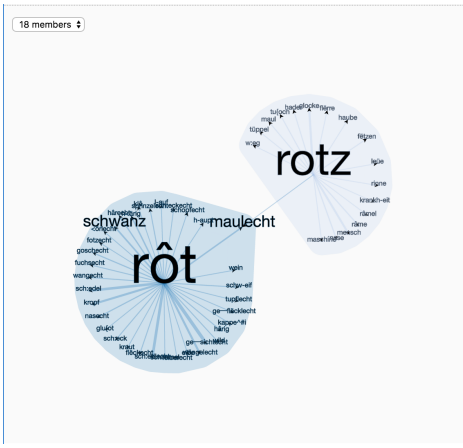


Figura 21: Detalle del grafo filtrando comunidades de menos de 18 elementos.

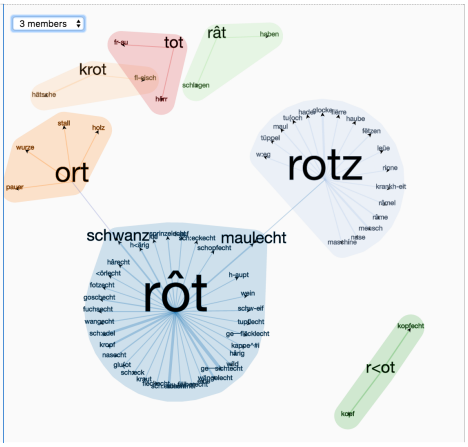


Figura 22: El mismo grafo mostrando ahora comunidades menos pobladas.

que el resto y que resultan de haber encajado en los términos de la búsqueda difusa (Ambas palabras tienen una distancia de un carácter con "rot"). El resto de nodos más pequeños representan las apariciones de otros lemas en las partes derechas de las mismas. Entre ellos destaca el lema "maulecht" que, a pesar de permanecer a la comunidad de "rôt", también se asocia con "rotz". Esto es indicador para la analista de un hecho curioso, que ha sido detectado gracias a métodos exclusivamente visuales. La analista decide seguir investigando este hecho modificando el selector del nivel de filtro para mostrar comunidades menos pobladas en un intento de encontrar más relaciones con el término "maulecht". El grafo se modificado de acuerdo a esta nueva selección y aparece como se refleja en la Figura 22.

Al aparecer más resultados pertenecientes a comunidades más pequeñas (los términos que encajaron en la búsqueda participan en menos palabras), se ven otros términos centrales, como "r<ot", que es una pronunciación diferente de la estándar "rot" y participa del término "pelirrojo" ("r<otkopf) y otras derivaciones sin relación con el color rojo.

5.3. Detalles en demanda después

Al no encontrar más relaciones, se continua el estudio analizando las palabras "rotzmaulecht" y "rôtmaulecht". Las preguntas que la analista se hace ahora se refieren a las particularidades de ambos términos: ¿Cumplen "rot" y "rotz" la misma función dentro de la palabra? ¿Expresan ambos lo mismo? Es por tanto que va a necesitar el detalle de las mismas. Desde el grafo se van a buscar ambos términos, primero uno y luego otro, llegando hasta el registro que los origina. Analizando el registro original TUSTEP, se va a poder comprobar que

“rôtmaulech” hace referencia al “sonrojo”, mientras que “rotzmaulecht” significa “estirado”, en sentido peyorativo. En este caso la analista ha descubierto que, a pesar de existir una relación entre términos parecidos a través de “maulecht”, estos no guardan ninguna relación en significado, al menos de manera aparente, dando lugar al fenómeno de la **homofonía**. Si la analista repite la búsqueda, esta vez centrándose en otras zonas geográficas de la Figura 19, puede encontrar el caso contrario (Ver Figura 23).

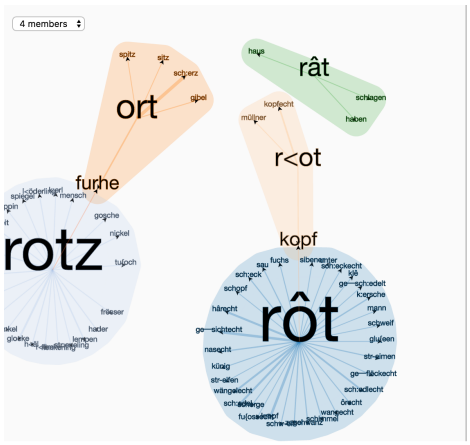


Figura 23: Un grafo en el que dos pronunciaciones de “rot” se asocian con el mismo lema “kopf” produciendo palabras con significados diferentes.

En el nuevo grafo de comunidades se ha identificado otra pronunciación de “rot”, vista en el ejemplo previo también: “r<ot”. Este término se asocia con “kopf”, que a su vez se relaciona con “rôt”. Si la usuaria repite el proceso y accede a los registros individuales de cada uno, podrá comprobar que “r<otkopf” se refiere a “Caperucita Roja”, mientras que “rôtkopf” hace referencia a la persona pelirroja. Este fenómeno se conoce como **homografía**. En este ejemplo concreto, ambas derivaciones del término de búsqueda textual empleado “rot” mantienen su significado (el color rojo) dentro del concepto al que definen, sin embargo en esta ocasión también derivan en términos con significados diferentes, deduciendo que el lema que experimenta la polisemia es en este caso “kopf”. Un estudio posterior más amplío podría por ejemplo determinar si esta manera de denominar a “Caperucita Roja” es particular de cierto dialecto (y de ahí la variación pronunciación) o si por el contrario ésta es la forma habitual en todos los territorios.

6. Conclusiones y líneas de trabajo futuras

6.1. Conclusiones

Como se comenzó diciendo al principio de esta memoria, la lexicografía es una rama sumamente complicada de las Humanidades en la que intervienen multitud de disciplinas y factores diferentes: históricos, políticos, sociológicos, antropológicos... Las Ciencias de la Computación tienen también una larga historia en su relación con esta materia y no es probable que esta tendencia termine en un futuro cercano. Existen aún multitud de campos en los que este tipo de colaboraciones entre las dos ciencias son muy escasas y se necesitará del esfuerzo de muchas investigadoras más para cubrir por completo estos campos del conocimiento humano.

En nuestra investigación tratamos por medio de una de estas colaboraciones crear un método y una herramienta de trabajo que consiguiesen aligerar el esfuerzo que tradicionalmente ha demandado el estudio de estas cuestiones. No obstante, la sensación final es que a pesar de los logros conseguidos y aún queda mucho camino por recorrer. Aunque cada día actores de ambas disciplinas encuentran puntos en común sobre los que realizar trabajo significativo, éstas siguen estando bastante alejadas tanto en discurso como en método. El perfil de un investigador en al área de las C. de la Computación sigue asociado con un tipo de mentalidad que dista mucho de aquella típica de un humanista y viceversa, y a pesar de la cada vez más rápida evolución de los avances tecnológicos y los sistemas educativos, los diálogos y el intercambio de ideas entre las partes siguen siendo complicados y en ocasiones incluso infructíferos.

En este trabajó se defendió el papel protagonista de la visualización en la liberación de la potencia de los métodos matemáticos y computacionales de los que se beneficia la lingüística, como los ya mencionados análisis de redes, procesamiento del lenguaje natural y sistemas de información geográfica. En este trabajo se vieron casos reales de puesta en práctica de muchos de los recursos ofrecidos por cada uno de ellos con el objetivo común de desempeñar una tarea específica de investigación bien definida. El cambio de paradigma propuesto con respecto a muchas de las soluciones de visualización de datos preexistentes se ha considerado uno de los logros más importantes de esta investigación. La propuesta arquitectónica sobre la que se asienta ha probado que hoy en día es posible visualizar adecuadamente grandes conjuntos de datos en el navegador empleando tecnologías y estándares web abiertos sin comprometer el nivel de rendimiento al que los usuarios están acostumbrados. A través de una metodología de desarrollo iterativa se pudo crear un sistema que no se desviase de los intereses del perfil investigador al que estaba orientado. Los diversos prototipos realizados consiguieron su propósito de dar una perspectiva diferente pero orientada hacia el mismo objetivo final de los conjuntos de datos, contribuyendo en mayor o menor medida a la consecución de los objetivos propuestos al principio de la investigación. En este proceso, y gracias a las distintas fases de prueba y reuniones con el equipo de expertas que potenció este modelo de desarrollo, se fue adquiriendo progresivamente una idea más detallada de la complejidad

de un problema que era en un principio totalmente ajeno al investigador y que redundó en una aplicación mejor orientada de las técnicas computacionales de las que hace gala el sistema final.

Las HD son un campo altamente interesante sobre el que aplicar de maneras novedosas soluciones a problemas clásicos de computación en contextos diferentes a los tradicionales. La necesidad de crear métodos y marcos de trabajo que dirijan adecuadamente esta experimentación se hace por tanto más patente que nunca ante el incipiente crecimiento tecnológico del que es partícipe la sociedad de hoy en día. El esfuerzo conjunto y coordinado de los diferentes grupos de investigación y otros actores involucrados en la puesta en práctica de esta colaboración, así como del resto de la sociedad es, por tanto, imprescindible. Para ello, el empleo de estándares abiertos que potencien el intercambio de datos, así como de nuevas técnicas de visualización preparadas para tratar con grandes flujos de información se convertirá en una necesidad cada vez más obvia en el día de mañana.

6.2. Líneas de trabajo futuras

Se detectaron durante la realización de esta investigación otras posibles líneas de trabajo que podrían asentarse en estas bases en el futuro como el **tratamiento visual de la incertidumbre** o las **búsquedas difusas**. Existen estudios[1][4] sobre el tratamiento visual de la incertidumbre: tanto la objetiva (entendida como falta de información, la imposibilidad física de apreciar), como la subjetiva (el estado mental del observante que denota falta de confianza en la información que se está recibiendo a través de los sentidos). En base a lo observado en otros trabajos mencionados en esta memoria, creemos que sería interesante realizar algún tipo de proyección de las estructuras de análisis de redes en el mapa. Los algoritmos de *pathfinding* en este contexto podrían ser de especial utilidad para encontrar distancias entre apariciones del mismo lema en puntos diferentes. Ejecuciones simultáneas de los mismos darían lugar a pistas visuales para encontrar coincidencias entre apariciones de los mismos términos en las mismas épocas o lugares. Si además se combinasen con animaciones en base a la dimensión temporal, servirían a la analista para buscar patrones de propagación de los lemas en base al tiempo y el espacio. La aplicación de algoritmos de inundación podría ser empleada para identificar "caminos" culturales por los que se transmitían los conceptos de unas partes a otras del mapa. La inclusión también de **fronteras geográficas históricas** para contextualizar esta información sería muy probablemente adecuada también.

En el grafo dirigido de fuerzas mostrado en el prototipo se aplica, como se comentó en la sección correspondiente, un algoritmo de detección de comunidades. El agrupamiento visual de elementos en los mismos en base a estas comunidades sería también interesante para crear visualizaciones más compactas en grafos superpoblados. Este tipo de implementaciones necesitarían de una mayor y más estrecha colaboración con el equipo de expertas para llevarlas a cabo debido a su gran complejidad. Por otra parte, la creación de grafos en base a otras características (relación de significado, de distancia, de tipo de palabra) también se

ha contemplado y se espera recibir más información al respecto cuando el equipo de expertas proceda en su investigación empleando el prototipo actual. El **enlace de elementos visuales del prototipo con copias digitales** provenientes de manuscritos originales también habrá de ser tenido en cuenta cuando estas colecciones adquieran mayor tamaño. Asimismo, aplicar enfoque de **Ciencia Ciudadana** en los que se integrase al conjunto de la población en partes específicas del flujo de trabajo de la investigación sería altamente beneficioso para completar y verificar la información de la que se dispone.

Referencias

1. BARTHELME, S.: Visual uncertainty (a bayesian approach) (2010)
2. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment 2008(10), P10008 (2008)
3. Busa, R.: The annals of humanities computing: The index thomisticus. Computers and the Humanities 14(2), 83–90 (1980)
4. Dasgupta, A., Chen, M., Kosara, R.: Conceptualizing visual uncertainty in parallel coordinates. In: Computer Graphics Forum. vol. 31, pp. 1015–1024. Wiley Online Library (2012)
5. Heeringa, W.J.: Measuring dialect pronunciation differences using Levenshtein distance. Ph.D. thesis, Citeseer (2004)
6. Mayer, T., List, J.M., Terhalle, A., Urban, M.: An interactive visualization of crosslinguistic colexification patterns. 09: 00–10: 30–Morning Session, Part I 09: 00–09: 10–Introduction 09: 10–09: 40 Thomas Mayer, Johann-Mattis List, Anselm Terhalle and Matthias Urban, An Interactive Visualization of Crosslinguistic Colexification Patterns 11(15), 1
7. Niemeyer, G.: Geohash (2008)
8. Orman, G.K., Labatut, V., Cherifi, H.: On accuracy of community structure discovery algorithms. arXiv preprint arXiv:1112.4134 (2011)
9. Ott, W.: Strategies and tools for textual scholarship: the tübingen system of text processing programs (tustep). Literary and linguistic computing 15(1), 93–108 (2000)
10. Theron, R., Fontanillo, L.: Diachronic-information visualization in historical dictionaries. Information Visualization 14(2), 111–136 (2015)
11. Theron Sanchez, R., Wandl-Vogt, E.: The fun of exploration: How to access a non-standard language corpus visually (2014)
12. Verbert, K.: On the use of visualization for the digital humanities, `http://dh2015.org/abstracts/xml/VERBERT_Katrien_On_the_Use_of_Visualization_for_t/VERBERT_Katrien_On_the_Use_of_Visualization_for_the_Dig.html`, ultimo acceso: 20-06-2016
13. Wandl-Vogt, E.: Point and find: the intuitive user experience in accessing spatially structured dialect dictionaries (2010)
14. Wanner, F., Jentner, W., Schreck, T., Stoffel, A., Sharalieva, L., Keim, D.A.: Integrated visual analysis of patterns in time series and text data-workflow and application to financial data analysis. Information Visualization 15(1), 75–90 (2016)

Sistema de minería de opiniones para el análisis de sentimiento en Twitter

Pamella Aquino, Vivian Batista

Universidad de Salamanca
{pabaquino,vivian}@usal.es

Resumen Tras el cambio de paradigma producido por la Web 2.0, el volumen de información subjetiva en Internet ha aumentado exponencialmente. Más allá de los sitios web, las redes sociales especialmente son una valiosa fuente de información puesto que unen gustos, preferencias y opiniones de usuarios de todo el mundo. Esta información es un desafío interesante desde la perspectiva del procesamiento del lenguaje natural (PLN). Este trabajo propone una combinación de métodos de análisis de sentimiento para alcanzar mejores resultados que los métodos existentes aplicados individualmente. La idea subyacente es el uso de técnicas de minería de datos y PLN para obtener automáticamente conocimiento útil acerca de las opiniones, preferencias y tendencias de los usuarios y poder hacer la clasificación de sentimientos y opiniones, sobre características de un producto o servicio. Combinando métodos clásicos de minería de textos, mediante un sistema de votación se elige la mejor clasificación para cada *tweet*, basada en la mayoría absoluta de los votos de los algoritmos considerados. El objetivo es determinar el sentimiento positivo o negativo, expresado en mensajes escritos en la red social *Twitter*.

1. Introducción

Tras el cambio de paradigma producido por la Web 2.0, el volumen de información subjetiva en Internet ha aumentado exponencialmente. El número de sitios en los que usuarios pueden expresarse, opinar, colaborar, crear, buscar y compartir contenido es cada vez mayor. Más allá de los sitios web, las redes sociales especialmente son una valiosa fuente de información puesto que unen gustos, preferencias y opiniones de una cantidad extraordinaria de usuarios de todas partes del mundo.

Dicha información es un desafío interesante desde la perspectiva del procesamiento del lenguaje natural (PLN), pero aparte de eso es un aspecto de profundo interés y enorme valor no sólo como estrategia de *marketing* para empresas y campañas políticas, sino también como indicador de medida de satisfacción de los consumidores de un producto o servicio.

Muchas áreas han tenido que reinventarse para lograr mantenerse en el mercado, ya sea cambiando la manera en que los negocios interactúan con sus clientes o la forma de promocionarse. Estar conectado a las redes sociales es apenas una de ellas. Otro mecanismo es aprovecharse de los datos proporcionados por los

usuarios, tratándolos y transformándolos en información, que puede ser utilizada para agregar valor a sus servicios. Debido a la inmensa cantidad de información, varias aplicaciones han surgido en un intento de extraer opiniones e incluso inferir el sentimiento público.

El reto de la interpretación de las opiniones y sentimientos expresados en forma textual dio lugar a la aparición de análisis de los sentimientos o minería de opiniones, que se remite al PLN, para identificar y extraer información subjetiva en el texto, con el fin de que se pueda clasificar como positiva o negativa. El análisis de sentimientos en redes sociales posibilita hacer una encuesta de gran alcance, no invasiva, rápida, auténtica, barata y automática [12], puesto que frecuentemente los usuarios proporcionan opiniones de forma espontánea sobre productos, servicios, eventos o marcas.

Existen diversos estudios, algoritmos y técnicas que se dedican a este tipo de análisis, pero ha sido recientemente que se ha aumentado el interés en adaptar estos algoritmos a textos cortos y con características peculiares. En parte, esto se debe al hecho de que las opiniones expresadas por los usuarios en las redes sociales tienen cada vez más importancia, como en el caso de *Twitter*[1], que delimita el espacio del mensaje a 140 caracteres.

La limitación de caracteres trasladada de los SMS a las redes sociales hizo que los usuarios crearan un código comunicativo propio a la hora de redactar sus mensajes, con el uso excesivo de abreviaturas, emoticonos para expresar su estado de ánimo, repetición de letras para dar énfasis o subir el tono e, incluso, errores ortográficos.

La propuesta de este trabajo es la combinación de métodos de análisis de sentimiento para alcanzar mejores resultados que los métodos existentes aplicados de forma individual. Para ello, se utiliza un corpus de datos de mensajes de *Twitter*, llamados de *tweets*, etiquetados con las clases positivo y negativo, el cual será entrenado mediante algoritmos supervisados, que clasificarán la polaridad del mensaje considerando una votación entre los algoritmos y determinará la polaridad final según la predicción de la etiqueta más popular. Además, se ha construido una herramienta de minería de opiniones, con una interfaz gráfica que ayude a los usuarios finales a observar la información útil encontrada y a poder visualizar el sentimiento.

El trabajo se encuentra estructurado de la siguiente forma: en la sección 2 se elabora una revisión del estado de arte de técnicas, métodos y trabajos relacionados; en la sección 3, se describen las etapas, tecnologías empleadas y el funcionamiento de la solución propuesta; en la sección 4, se muestran los experimentos realizados y los resultados obtenidos; en la sección 5, se introduce el prototipo de visualización. Por último, en la sección 6, se sintetizan las conclusiones sobre el trabajo y las líneas futuras de investigación.

[1] https://twitter.com/

2. Estado del Arte

El lenguaje natural es la principal herramienta utilizada de manera cotidiana por los humanos para expresarse, comunicarse con otras personas, transmitir conocimientos, sentimientos y emociones. Con el avance de las tecnologías y la expansión de la Internet, la cantidad de información generada en forma de lenguaje natural, proveniente principalmente de los nuevos medios digitales, ha crecido de manera vertiginosa, iniciando una era marcada por un volumen inmenso de información.

Sin embargo, para que esa información se transforme en conocimiento, los documentos almacenados en su mayor parte en formato electrónico, deben ser tratados, procesados y utilizados como fuente de información desde donde se puede buscar respuestas a preguntas, hacer inferencias lógicas respecto a su contenido, generalizar y resumir información, etc.

El desarrollo de la ciencia en los últimos años ha dado lugar a que todo el procesamiento de ese enorme volumen de información ocurriese, lo que trajo como consecuencia la creación de un nuevo campo de estudio conocido como PLN, que radica de la unión de los campos de la ciencia de la computación, inteligencia artificial y lingüística aplicada, con el objetivo de estudiar los problemas originados de la generación, comprensión y procesamiento automático del lenguaje natural y hacer posible la interacción y comunicación entre personas y máquinas.

Esta ciencia investiga cómo el lenguaje puede usarse para llevar a cabo diferentes tareas y cómo modelar el conocimiento. Pero dada la complejidad de la representación del lenguaje natural, algunas de sus propiedades reducen la efectividad de los sistemas de recuperación de información de textos.

En ese sentido, el preprocesamiento de los textos juega un papel muy importante en el PLN. Es más, debe ser siempre el primer paso hacia el proceso de clasificación, puesto que elimina atributos (*features*) irrelevantes o redundantes del texto. La normalización, tokenización, eliminación de palabras vacías, etiquetado gramatical son algunas de esas técnicas.

La clasificación de textos permite asignar una categoría de forma automática a un texto, ya sea un documento, expresión o sentencia. Con este fin, se apoyan en métodos conocidos como aprendizaje automático supervisado, que se valen de un conjunto previamente anotado, para entrenar sus algoritmos y ser capaces de extraer características importantes para las clasificaciones futuras. Los modelos de Bayes ingenuo (NB), y sus variantes, MultinomialNB y BernoulliNB, Máquinas de Vectores de Soporte (SVM), Modelo de Máxima Entropía (MaxEnt) pueden ser muy eficaces para solucionar problemas de clasificación y de análisis de sentimientos.

Los primeros trabajos realizados utilizando clasificación automática tuvieron lugar en 2002 por los autores Pang et al. [14] y Turney [21], que han usado métodos de aprendizaje supervisados y no supervisados en la clasificación de reseñas de películas y productos, respectivamente. Estos trabajos se han tomado como punto de partida para muchas obras en los años posteriores.

Estudios más recientes, como [11,2,3], han analizado publicaciones de textos cortos, que pese al ruido en el texto, como el uso de dialectalismos, coloquialis-

mos, ironias, ambigüedades, ortografía relajada y el uso exagerado de abreviaturas, han conseguido alcanzar buenos resultados.

Investigaciones realizadas en el campo de análisis de sentimiento han utilizado métodos léxicos (diccionarios de polaridad) [8][15][5][24], que permiten mejoras como, por ejemplo, el uso de conocimientos lingüísticos. Otras hacen una combinación de métodos para alcanzar mejores resultados de los obtenidos por enfoques individuales [15][1][22][7], y efectivamente lo están logrando. El resultado de cada método se combina en una única salida sacando mejores medidas de precisión y *F-measure* comparados a los métodos usados de forma individual. [22][7] han ido más allá, combinando los resultados de los métodos individuales por mayoría de votos mediante tres técnicas: votación simple, *bagging*[4,16] y *boosting*[25,19].

3. Metodología

La herramienta propuesta en este trabajo tiene como objetivo crear un modelo para extraer y clasificar de forma automática información haciendo uso y combinación de técnicas de Inteligencia Artificial. Para ello, se han dividido en siete de etapas:

1. Construcción de un corpus de datos
2. Preprocesamiento
3. Extracción de *features*
4. Clasificación
5. Evaluación
6. Análisis de Resultados
7. Visualización

Las tecnologías empleadas en el desarrollo de este trabajo se describen de manera breve a continuación: el lenguaje de programación *Python*, la plataforma de desarrollo NLTK[2] (*Natural Language Toolkit*) para el procesamiento del lenguaje natural, la biblioteca *Scikit-Learn*[3] para agregar algoritmos de aprendizaje automático y métricas de rendimiento, la biblioteca *Tweepy*[4] para interacción con la API de *Twitter*, la plataforma *NodeJS* para la creación de la visualización y, por último, la base de datos *MySQL*[5] para almacenar el conjunto de datos de la herramienta.

El objetivo de este trabajo es proponer un enfoque diferenciado a la tarea de análisis de sentimiento, combinando los métodos clásicos de clasificación de textos encontrados en la literatura, mediante un sistema de votación que elige la mejor clasificación para cada *tweet* basada en la mayoría absoluta de los votos de los algoritmos.

[2] http://www.nltk.org
[3] http://scikit-learn.org/
[4] http://www.tweepy.org/
[5] http://www.mysql.com/

Para el desarrollo del sistema de votación para la clasificación de los *tweets*, se ha decidido tomar como referencia el algoritmo utilizado por Harrison Kinsley en su plataforma Python Programming[6].

El sistema de votación está compuesto por cinco algoritmos clásicos (Naïve Bayes, MultinomialNB, BernoulliBN, SVM y MaxEnt), donde cada uno tiene derecho a un voto. Tras la votación, la clase con la mayoría de los votos se asigna como la clase del *tweet*.

Para alcanzar la mayoría absoluta en este caso, es necesario que por lo menos tres votos coincidan. Ese umbral mínimo estima un grado de confianza del 60 % para obtener la solución correcta. Si cuatro de los algoritmos votan a la misma clase, el grado de confianza aumenta en un 80 %, y si los cinco coinciden, el grado de confianza alcanza el 100 %.

Es importante resaltar que el grado de confianza del 100 % no significa que la clasificación ha sido totalmente precisa. Hay que tener en cuenta que la precisión más alta alcanzada por los cinco algoritmos a partir del conjunto de entrenamiento provisto ha sido del 78.8 %, con el método de extracción de frecuencia de palabras. Sin embargo, a diferencia de los métodos abordados en el estado del arte, en lugar de considerar solamente un algoritmo, se tiene el aval de cinco de ellos para garantizar que la clasificación de la clase sea correcta.

La precisión del sistema se calcula en base al promedio de la métrica de *accuracy* de todos los algoritmos involucrados en el sistema de votación, siendo esta el 78.06 % (véase la sección 4).

4. Experimentación y Resultados

Los experimentos realizados se llevan a cabo mediante las siete etapas descritas en la sección anterior. Las cuales se describen con de forma resumida a continuación.

1. **Construcción de un corpus de datos**
   Para la construcción de un corpus sólido y bien clasificado se utilizó la herramienta *Tweepy* aplicando el método de recogida de datos empleado por [17][10], basado en supervisión a distancia (*distant supervision*) [9], haciendo filtrado de *tweets* empleando emoticonos según la polaridad. El proceso de etiquetado automático consistió en suponer que los *tweets* con los emoticonos positivos como ":)" asumirán una polaridad positiva y que los emoticonos negativos como ":(" asumirán una polaridad negativa. Sin embargo, hay múltiples emoticonos que pueden expresar emociones positivas, como por ejemplo ":-)" y ":]", y emociones negativas, como ":-(" y ":[". Una lista más completa se puede ver la tabla 1.
   Otro parámetro determinado en el filtrado, ha sido el idioma, los *tweets* recuperados están escritos en inglés. Por lo tanto, la clasificación propuesta sólo funcionará con *tweets* en inglés porque los datos para el entrenamiento están en ese idioma.

[6] https://pythonprogramming.net/

Tabla 1: Lista de emoticonos

Positivos	:) :-) :] :> :} :o) :c) >:] ;) =] =) 8) :p :P =D :o :D :-D :)
Negativos	:(:-(:[:< :{ :o(:c(>:[;(=[=(8(:3 :/

Al corpus se ha aplicado un postprocesado eliminando los siguientes elementos: emoticonos (listados en la tabla 1), *tweets* repetidos, *retweets*[7], *tweets* con menos de 5 *tokens*.
Tras el postprocesado se han obtenido un total de 18.000 *tweets*, siendo 9.000 con sentimiento positivo y 9.000 con sentimiento negativo, que se almacenan en dos ficheros diferentes, según la polaridad. El periodo de recogida del corpus fue del 3 de junio hasta 5 de junio de 2016.

2. **Preprocesamiento**
   Sumado al preprocesamiento clásico de las técnicas de PLN, se realiza una segunda etapa de preprocesamiento. En esta ocasión, para tratar las construcciones específicas del modelo de lenguaje utilizado en *Twitter*, que no aportan información a la tarea de clasificación, como los nombres de usuarios, enlaces, repetición de letras y sustitución de *hashtag* por el sintagma que lo compone.
   El espacio de datos provisto por el corpus originalmente suministraba 38.138 *features* y tras la tarea de preprocesamiento hubo una reducción de dimensionalidad del 53.02 %, restando apenas 17.915 *features*.

3. **Extracción de *features***
   Los procesos de extracción de *features* empleados se basan en los modelos de lenguaje bolsa de palabras [20], considerando la frecuencia de palabras y el *PoS Tagging* [13] con la frecuencia de adjetivos.
   En la frecuencia de palabras, se consideran *features* todos los términos del conjunto de entrenamiento, sin importar la posición, y en cada documento se tiene en cuenta la frecuencia con la que ocurre cada palabra según la polaridad. En este caso, se ha considerado el espacio de datos de 17.915 *features*, tras el procesamiento realizado en la etapa anterior.
   Por otra parte, en el *PoS Tagging* con la frecuencia de adjetivos, se consideran *features* todos los adjetivos del conjunto de entrenamiento y en cada documento se tiene en cuenta la frecuencia con la que ocurre cada adjetivo, según la polaridad. En este caso, el espacio de datos se limita solamente a los adjetivos, que corresponden al 19.8 % del total de datos, es decir, 3.545 *features*.

[7] *Retweet* es una forma de republicar un *tweet* de algún usuario que se sigue o que considera interesante y normalmente se identifican por la abreviación “RT’.

4. **Clasificación**
   El proceso de clasificación adoptado se basa en el aprendizaje supervisado para predecir la polaridad (positiva o negativa) de cada *tweet*. Para ello, se utiliza el conjunto de entrenamiento correspondiente a 75 % del corpus de datos etiquetado, con 13.500 *tweets*, para entrenar los cinco clasificadores provistos por el sistema de votación. Tras el entrenamiento, se evalúan los clasificadores frente al conjunto de prueba (4.500 *tweets*) para obtener su rendimiento. En esta etapa, se ha empleado la implementación de los algoritmos suministrada por las bibliotecas NLTK y *Scikit-Learn*.

5. **Evaluación**
   Para la evaluación de los clasificadores, se han considerado la implementación suministrada por la biblioteca *Scikit-Learn* de las métricas *accuracy, precision, recall* y *F-measure* [18], para las clases positivas y negativas, con relación a los métodos de extracción de frecuencia de palabras y el *PoS Tagging* con adjetivos.

Tabla 2: Rendimiento de los Clasificadores - Frecuencia de Palabras

Clasificador	$Accuracy$	$Precision_{pos/neg}$	$Recall_{pos/neg}$	$F-measure_{pos/neg}$
Naïve Bayes	78.33	91.7/87	86.4/92.1	89/89.5
MultinomialNB	78.8	91.5/84.9	83.8/92.1	87.5/88.4
BernoulliNB	78.8	90.4/85.5	84.8/91	87.5/88.1
MaxEnt	78.6	91.9/91	91/91.9	91.5/91.4
SVM	75.8	97.4/96.9	96.9/97.3	97.1/97.1

Tabla 3: Rendimiento de los Clasificadores - *PoS Tagging* con Adjetivos

Clasificador	$Accuracy$	$Precision_{pos/neg}$	$Recall_{pos/neg}$	$F-measure_{pos/neg}$
Naïve Bayes	69.86	76/78.8	80/74.6	78/76.7
MultinomialNB	70.07	76.2/78	78.8/75.3	77.5/76.6
BernoulliNB	69.93	75.6/78.2	79.5/74.1	77.5/76.1
MaxEnt	70.31	75.8/80	81.6/73.7	78.6/76.7
SVM	69.42	77.7/82.3	83.8/75.8	80.6/78.9

En la figura 1 y la figura 2 se introducen los valores de *accuracy* obtenidos para cada algoritmo en función del número de *features* utilizado en el entrenamiento de los clasificadores, presentados en gráficos por cada uno de los

métodos de extracción de *features*.

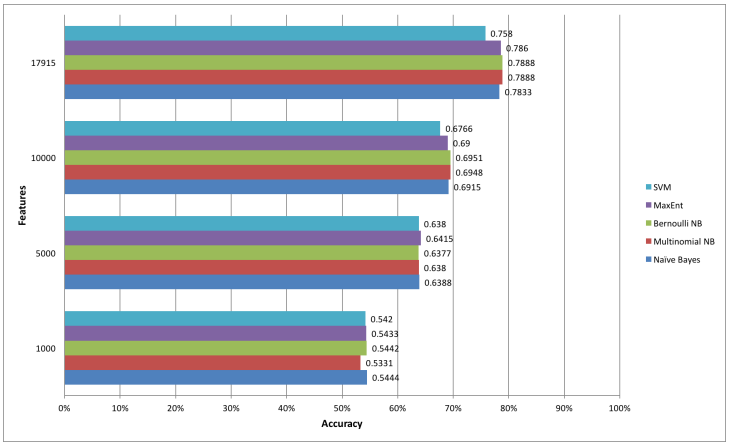


Figura 1: Rendimiento de Clasificadores por Frecuencia de Palabras: *Accuracy* × *Número de Features*

Tabla 4: Rendimiento de Clasificadores por Frecuencia de Palabras: *Accuracy* × *Número de Features*

Clasificador	1.000	5.000	10.000	17.915	Promedio
Naïve Bayes	54.44	63.88	69.15	78.33	66.45
MultinomialNB	53.31	63.8	69.48	78.88	66.37
BernoulliNB	54.42	63.77	69.51	78.88	66.65
MaxEnt	54.33	64.15	69	78.6	66.52
SVM	54.2	63.8	67.66	75.8	65.37

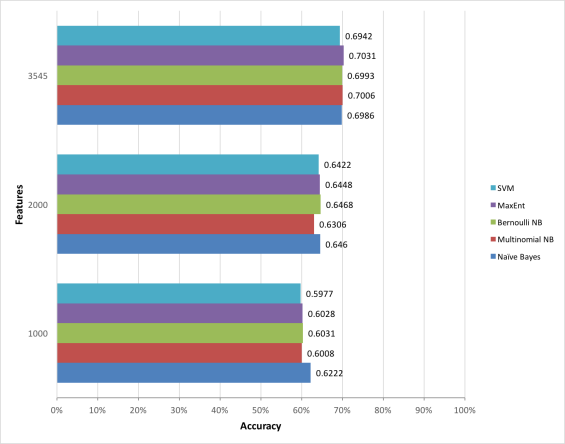


Figura 2: Rendimiento de Clasificadores por *PoS Tagging* con adjetivos: *Accuracy* × *Número de Features*

Tabla 5: Rendimiento de Clasificadores por *PoS Tagging* con adjetivos: *Accuracy* × *Número de Features*

Clasificador	1.000	2.000	3.545	Promedio
Naïve Bayes	62.22	64.6	69.86	65.56
MultinomialNB	60.08	63.06	70.06	64.4
BernoulliNB	60.31	64.68	69.93	64.97
MaxEnt	60.28	64.48	70.31	65.02
SVM	59.77	64.22	69.42	64.47

6. **Análisis de Resultados**

Analizando los resultados obtenidos del rendimiento de los clasificadores, se puede identificar que, para el proceso de extracción de *features* mediante frecuencia de palabras (Tabla 2), cuatro de los cinco algoritmos tuvieron medidas de *accuracy* muy similares, por encima del 78 %. El algoritmo SVM se destacó de los demás clasificadores no sólo en relación a la métrica *F-measure*, que evalúa el clasificador con una medida única, con un valor del 97.1 %, sino también con respecto a las demás métricas.

Del mismo modo, los resultados del rendimiento de los clasificadores mediante *PoS Tagging* con adjetivos (Tabla 3) tuvieron medidas de *accuracy* muy próximas, en torno al 70 %. Además, como en el caso anterior, el clasificador SVM sobresalió no sólo por la métrica *F-measure*, sino también por las demás métricas.

Sumado a lo anterior, también se ha analizado el rendimiento de los algoritmos en función del número de *features* utilizado en el entrenamiento de

los clasificadores (Figuras 1 y 2). El hecho de entrenar el clasificador con pocas *features* acarrea en un rendimiento más bajo. Sin embargo, a medida que aumenta el número de *features*, mejora el rendimiento del clasificador. Otro punto observado es el hecho de que el rendimiento de los algoritmos es muy parecido para la misma cantidad de *features* que se empleen en el entrenamiento.

En las tablas 4 y 5, se resaltan en amarillo las celdas en las que el clasificador ha obtenido mejor rendimiento en función del número de *features* para cada método de extracción. A su vez, la última columna muestra el promedio del rendimiento de cada clasificador. Para el método de frecuencia de palabras, el algoritmo Bernoulli destaca de los demás por obtener mejor rendimiento medio en función de la cantidad de *features*. En cambio, para el *PoS Tagging* con adjetivos, Naïve Bayes es el que tiene mejor comportamiento comparado a los demás clasificadores.

7. **Visualización**
   La visualización se introduce en el prototipo de la herramienta en la siguiente sección.

5. Prototipo de la herramienta

El prototipo de la herramienta desarrollada propone analizar cómo caso de estudio los datos publicados en *Twitter* con objeto de entender el comportamiento de los usuarios y el sentimiento de estos con relación a un producto o servicio. Se ha utilizado el enfoque propuesto en este trabajo, para identificar el sentimiento, positivo o negativo, expresado por los *tweets* publicados.

En ese contexto, se ha elegido analizar los datos del servicio de la *startup* americana *Uber*[8], que es una empresa que proporciona un servicio de transporte alternativo que ha cambiado la industria del taxi. Con un servicio que atiende en los cinco continentes, su entrada en el mercado ha provocado una reacción virulenta que se ha reflejado en las redes sociales. Muchos de los cambios fueron positivos, sin embargo, todo servicio, tiene sus fallos y aciertos. Gran parte de las disconformidades, quejas, denuncias o sugerencias sobre el servicio los usuarios las exponen en los mensajes en *Twitter*.

Los datos fueron recolectados durante el periodo comprendido entre el 20 de junio y 4 de julio de 2016. Haciendo filtrado por la palabra clave "*uber*", en total se recogieron 9.590 *tweets*, que se almacenaron en una base de datos *MySQL*.

El objetivo de la visualización en esta herramienta es permitir al analista de forma intuitiva, identificar la distribución de los *tweets* en un mapa del mundo y poder analizarlos. Los *tweets* se ubican en el mapa a partir de la localización provista por el usuario.

[8] https://www.uber.com/

El empleo de un mapa con la distribución de *tweets* va a permitir al analista[9] visualizar el sentimiento de los consumidores en relación al servicio, haciendo que la visualización se utilice como soporte en la toma de decisiones, porque desde ahí se pueden identificar sus fortalezas y debilidades y reconocer nuevas amenazas y oportunidades.

La herramienta está disponible en la dirección `http://104.131.53.43:2001/`, desde donde se puede interactuar y probar sus funcionalidades.

La estrategia de navegación empleada consiste en dos vistas: una global y otra detallada. En la vista global, se plantea un mapa del mundo que imprime un mapa de calor, que representa la intensidad de los datos indicando los puntos geográficos donde se ha hablado de Uber (a la derecha la pantalla en la figura 3). Por otra parte, el detalle se logra mediante la selección de una ciudad o área en el mapa, que recupera y enseña los *tweets* de la región seleccionada en una tabla coloreada en verde y rojo, destacando el sentimiento positivo y negativo, respectivamente, de cada mensaje, como se puede observar a la izquierda de la pantalla en la figura 3.

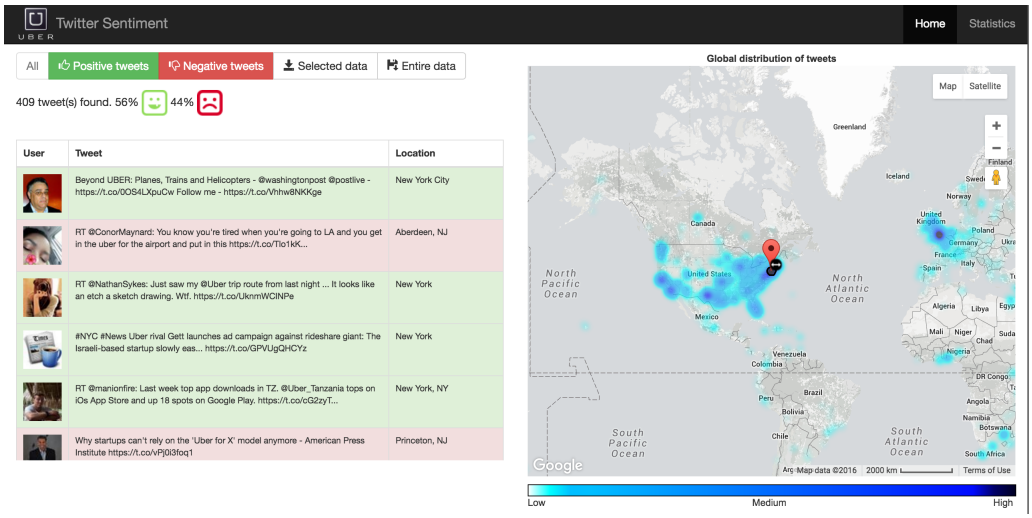


Figura 3: Visualización de vista global de la herramienta

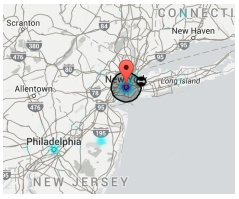
El mapa de calor resalta mediante un código de colores zonas concretas donde los usuarios han publicado sus mensajes indicando su localización. Esta representación utiliza la técnica de "*eye tracking*" [6] para ayudar al analista a detectar dónde se debe posar la mirada y así encontrar de forma intuitiva donde efecti-

[9] Se ha utilizado el término *analista* para referirse al usuario que maneja la herramienta y el término *usuario* para referirse al usuario de *Twitter* que publica mensajes en la red social.

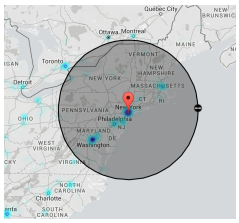
vamente hay datos para el análisis. Los colores empleados en el mapa de calor indican a través de una escala de intensidad que varía de azul bien claro a azul oscuro, según una menor o mayor frecuencia de *tweets*.

El enfoque interactivo planteado en la herramienta permite la manipulación directa del analista con varios elementos de la visualización: como el marcador, la tabla y los botones.

El *marcador* consiste en un icono que identifica una ubicación en un mapa. Este está diseñado para ser interactivo, es decir, se permite que los analistas lo muevan en el mapa fijándolo en un punto deseado, por ejemplo una ciudad (Figura 4a), y/o aumenten su área de cobertura pinchando y arrastrando el radio del marcador, haciéndolo más grande o más pequeño, seleccionando una región según su interés (Figura 4b).



(a) Marcador ubicado en una ciudad



(b) Marcador con área de cobertura

Figura 4: Usos del marcador en el mapa

La *tabla* mostrada en la figura 5, cuyo contenido son *tweets* de la base de datos, contiene 3 atributos: *User*, *Tweet*, *Location*, que serán descritos a continuación.

- **User**: Muestra la foto del usuario, permite al analista hacer clic y visualizar en la página de *Twitter* el perfil del usuario que publicó el *tweet*.
- **Tweet**: Muestra el mensaje publicado.
- **Location**: Señala la localización informada por el usuario.

El sentimiento de los *tweets* se indica por el color de la fila, que varia entre rojo y verde, donde rojo indica un *tweet* con sentimiento negativo y el verde, un *tweet* con sentimiento positivo.

Tras mover el marcador y determinar el área demarcada, la tabla con los *tweets* se retroalimenta de forma inmediata, actualizando los *tweets* según el rango de los límites señalados. En la parte superior de la tabla se indica el número de *tweets* recuperados y el porcentaje de *tweets* positivos y negativos de la muestra seleccionada.

Otra forma de interacción del analista con la herramienta es a través de los botones dispuesto por encima de la tabla, ilustrados en la figura 5. Cada uno tiene una funcionalidad específica.

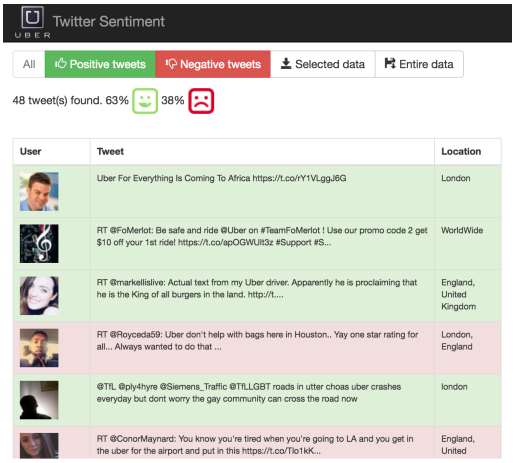


Figura 5: Visualización de vista detalle de la herramienta

Los botones "Positive Tweets" y "Negative Tweets" actualizan la tabla y plotean en el mapa solamente los *tweets* positivos o solamente los *tweets* negativos, respectivamente, según la selección del usuario. El analista puede mover el marcador a la región deseada y visualizar los *tweets* de apenas uno de los sentimientos. El botón "All" restaura los cambios aplicados por los otros dos botones, actualizando el mapa y volviendo a enseñar todos los *tweets*. Como la base de datos está bastante balanceada, el 45 % positivo y el 55 % negativo, y bien distribuida alrededor del mundo, cuando se pulsan los botones "Positive Tweets" y "Negative Tweets" no hay un gran impacto en el mapa, pero si se observa en las áreas de más intensidad, sí se puede percibir los cambios.

Por último, los analistas también pueden interactuar con los botones "Selected Data" y "Entire Data", cuyas funciones se describen a continuación.

- **Botón "Selected Data":** Permite descargar los *tweets* mostrados en la tabla, que corresponden a los datos del área seleccionada, en un fichero en el formato CSV.
- **Botón "Entire Data":** Permite descargar todos los *tweets* de la base de datos, también en ese formato.

Uno de los puntos débiles de la visualización es el hecho de que el mapa de calor no consigue reflejar el sentimiento de los *tweets*. En realidad, el empleo del mapa de calor en este trabajo se debe a que el servicio de *Uber* no está disponible en todos los sitios del mundo, así que fue usado para auxiliar al analista a la hora de mover el marcador a donde efectivamente hayan *tweets*.

La herramienta también proporciona la visualización de estadísticas de los datos recogidos. Estos datos se pueden acceder a partir de la pestaña "*Statistics*" localizada en la esquina superior derecha de la pantalla, como se muestra en la figura 6.

Los datos estadísticos provistos, mostrados en la figura 6, incluyen: los cinco países y ciudades donde más se publicaron *tweets*, la proporción de *tweets* válidos positivos y negativos y la proporción entre *tweets* válidos y *tweets* descartados. Se han considerado *tweets* válidos aquellos recuperados con el atributo "Location" no vacío y que, a partir del cual, fue posible transformar el lugar informado en coordenadas geográfica. Algunos de los lugares informados por los usuarios que no fue posible geolocalizar son "*My desk*", "*Working*", "*Easy to find, hard to predict*", por poner unos ejemplos.

Figura 6: Visualización de la pestaña *Statistics*

6. Conclusiones y Líneas Futuras de Investigación

En este trabajo se ha aplicado un nuevo enfoque al análisis de sentimiento. La idea subyacente es el uso de técnicas de minería de datos y procesamiento de lenguaje natural para obtener, de forma automática, conocimiento útil acerca de las opiniones, preferencias y tendencias de los usuarios y poder hacer la clasificación de sentimientos y opiniones, sobre características de un producto o servicio.

Combinando métodos clásicos de minería de textos, mediante un sistema de votación se elige la mejor clasificación para cada *tweet*, basada en la mayoría absoluta de los votos de los algoritmos considerados. El objetivo es determinar el sentimiento positivo o negativo, expresado en mensajes escritos en la red social *Twitter*.

La investigación ha permitido profundizar y comparar sistemas de minería de opiniones basados en redes sociales, a partir de técnicas de procesamiento del lenguaje natural y minería de datos. Alcanzándose el objetivo propuesto de identificar y extraer de forma automática, información subjetiva como opiniones, sentimientos y emociones de los usuarios. Para almacenarla de forma estructurada, poder procesarla y clasificarla como información útil.

Por lo que también se ha alcanzado otro objetivo más específico, construir una herramienta de minería de opiniones, que combine de forma automática dichos algoritmos de clasificación con sus correspondientes fases de selección de atributos, a través del preprocesamientos y postprocesamiento de textos. En este sentido se ha constatado la importancia de construir un corpus de datos bien etiquetado, con el fin de tener un conjunto de datos robusto para crear el modelo. El método en la recolección de datos [17] ha sido muy efectivo, puesto que sería muy difícil extraer suficientes datos manualmente.

Se ha verificado que la tarea de procesamiento es esencial para el buen rendimiento del sistema, ya que elimina todos los términos que no aportan valor a la clasificación y mejora la precisión de los resultados. Pudiéndose reducir el espacio de datos a menos de la mitad del conjunto original.

También se han evaluado, de forma independiente, cada uno de los clasificadores para verificar su rendimiento. Al analizar los resultados se ha notado una gran similitud entre ellos. Por ese motivo, se ha decidido plantear una solución combinada de los algoritmos, mediante un sistema de votación por mayoría, con el objeto de aumentar la confianza de la clasificación de los *tweets*.

Finalmente, todo ese proceso se plasma en una herramienta de visualización del prototipo propuesto, el cual ofrece una interfaz intuitiva, de fácil uso y enfocado en asistir al usuario final en la toma de decisiones. Permitiéndole encontrar patrones, mediante el mapa de distribución *tweets* y acceder a esos datos. Como caso de estudio del prototipo, se ha utilizado *Uber* para ejemplificar el uso de la herramienta con el objetivo de identificar, a través de los *tweets* positivos, oportunidades de seguir mejorando su servicio, fortalecer su marca y fidelizar a sus clientes. Mediante los *tweets* negativos, reconocer sus puntos de mejora y ayudar a resolver las quejas de sus clientes.

Como líneas futuras de investigación, se propone estudiar soluciones para resolver el problema de ambigüedad semántica utilizando conjuntos de entrenamiento en dominios específicos para mejorar los resultados de los clasificadores, evaluar otros tipos de preprocesamientos de texto, como por ejemplo, la corrección ortográfica, y mejorar la interfaz propuesta con la identificación del sentimiento en el mapa.

Referencias

1. Augustyniak, L., Kajdanowicz, T., Szymanski, P., Tuliglowicz, W., Kazienko, P., Alhajj, R. y Szymanski, B. *Simpler is better? lexicon-based ensemble sentiment classification beats supervised methods.* In: IEEE. Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on. [S.l.]. p. 924-929. 2014.
2. Bermingham, A. y Smeaton, A. F. *Classifying sentiment in microblogs: is brevity an advantage?*. En: ACM. Proceedings of the 19th ACM international conference on Information andknowledgemanagement.[S.l.] p.1833-1836. 2010.
3. Bermingham, A. y Smeaton, A. *On using Twitter to monitor political sentiment and predict election results.* En: Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2011). 2011.

4. Breiman, L. *Bagging predictors.* Machine Learning 24 (2) 123-140. 1996.
5. Choy, M., Cheong, M. L. F., Nang Laik, M. y Ping Shung, K.. *A sentiment analysis of Singapore Presidential Election 2011 using Twitter data with census correction.* 2011. CoRR, abs/1108. 5520.
6. Duchowski, A. T. *A Breadth-First Survey of Eye Tracking Applications.* Behavior Research Methods, Instruments & Computers (BRMIC), 34(4), pp. 455-470. November 2002.
7. Fersini, E., Messina, E. y Pozzi, F. *Sentiment analysis: Bayesian ensemble learning.* Decision Support Systems, Elsevier, v.68, p.26-38. 2014.
8. Hu, M. y Liu, B. *Mining and summarizing customer reviews.* En: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, pages 168-177, New York, NY, USA. ACM. 2004.
9. Intxaurrondo, A., Surdeanu, M., López de Lacalle Lekuona, O. y Agirre Bengoa, E. *Removing noisy mentions for distant supervision.* Procesamiento del Lenguaje Natural. N. 51 . ISSN 1135-5948, p. 41-48. 2013.
10. Go, A., Huang, L. y Bhayani, R. *Twitter sentiment analysis.* Final Projects from CS224N for Spring 2008/2009 at The Stanford Natural Language Processing Group. 2009.
11. Kouloumpis, E., Wilson, T. y Moore, J. *Twitter sentiment analysis: The good the bad and the omg!* Icwsm, v.11, p.538-541, 2011.
12. Malheiros, Y. y Lima, G. *Uma Ferramenta para Análise de Sentimentos em Redes Sociais Utilizando o SenticNet.* Simpósio Brasileiro de Sistemas de Informação, IX, p. 517-522, 2013.
13. Manning, C.D y Schuetze, H. *Foundations of statistical language processing.* Cambridge, MA: MIT Press, 1999.
14. Pang, B., Lee, L. y Vaithyanathan, S. *Thumbs up? Sentiment classification using machine learning techniques.* En: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), p. 79-86, 2002.
15. Prabowo, R. y Thelwall, M. *Sentiment analysis: A combined approach.* Journal of Informetrics, v.3, n.2, p.143-157. 2009.
16. Quinlan, J.R. *Bagging, boosting, and c4.5.* Proceedings of the 13th National Conference on Artificial Intelligence, pp. 725-730. 1996.
17. Read, J. *Using emoticons to reduce dependency in machine learning techniques for sentiment classification.* En: Proceedings of ACL-05, 43nd Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2005.
18. Salton, G. y McGill, M.J. *Introduction to Modern Information Retrieval.* New York, NY, USA: McGraw-Hill, Inc.; 1986.
19. Schapire, R.E. *The strength of weak learnability.* Machine Learning 5 (2) 197-227. 1990.
20. Sebastiani, F. *Machine Learning in Automated Text Categorization.* ACM Computing Surveys, 34(1), pp. 1-47. 2002.
21. Turney, P. *Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews.* En: Proceedings of the Association for Computational Linguistics (ACL), pages 417-424, 2002.
22. Wang, G., Sunc, J., Mac, J., Xue, K. y Gud, J. *Sentiment classification: The contribution of ensemble learning.* Decision Support Systems, Elsevier, v.57, p.77-93, 2014.
23. Wilson, T., Wiebe, J. y Hoffmann, P. *Recognizing contextual polarity in phrase level sentiment analysis.* En: Proceedings of the Conference on Human Language

Technology and Empirical Methods in Natural Language Processing, HLT '05, p. 347-354, Stroudsburg, PA, USA. ACL. 2005.

24. Zhang, L., Ghosh, R., Dekhil, M., Hsu, M.; y Liu, B. *Combining Lexicon-based and Learning-based Methods for Twitter Sentiment Analysis.* Hewlett-Packard Laboratories. Technical Report HPL-2011-89. 2011.
25. Zhou, Z.-H. *Ensemble Methods: Foundations and Algorithms* Chapman & Hall, 2012.

Análisis del daño que provoca el vandalismo en la Wikipedia

Gerardo Andres Corado Juárez y Angel Zazo Rodríguez

Universidad de Salamanca
Departamento de Informática y Automática
Facultad de Ciencias
Plaza de los Caídos s/n
37008 Salamanca, Spain

gerardo.corado@usal.es, zazo@usal.es,

Resumen Por la naturaleza colaborativa de la Wikipedia existen editores que alteran de forma incorrecta la información de los artículos (por desconocimiento o por producir daño) provocando vandalismo; esta información alterada es visualizada por un número de usuarios en el tiempo que dura el contenido. La visualización puede generar problemas para las personas que desean consultar los artículos afectados. Por lo que en este trabajo se estudiará el impacto o el número de consultas afectadas por el vandalismo, par ello se utiliza una herramienta visual que ilustra el efecto del vandalismo sobre un año de edición del artículo. También se realiza un análisis del daño que provoca el vandalismo en un mes identificando los artículos más afectados y quienes son los editores más vándalos y los mas rever-sores utilizando una fórmula para medir los accesos de un fragmento de tiempo.

1. Introducción

Colaborar siempre ha sido parte fundamental para alcanzar objetivos dentro la sociedad humana. La comunicación moderna ha facilitado la creación de nuevas y excitantes formas de cooperación donde los colaboradores interactúan remotamente con una velocidad increíble intercambiando una basta cantidad de información. El ejemplo más famoso es la World Wide Web que fue creada con el objetivo de tener una plataforma apropiada para colaboraciones masivas. Wikipedia es un ejemplo de ambiente colaborativo que tiene la ventaja que todas sus interacciones están bien documentas y disponibles públicamente siendo apropiada para estudios científicos; es de las páginas más visitadas en Internet, datos de Alexa[1], en que cada individuo puede modificar el contenido de cualquier página. La misma Wikipedia se describe como una enciclopedia libre políglota donde cualquier lector puede ser autor de ella; pero también es su maldición porque hay usuarios que abusan de su naturaleza libre alterando los artículos con el fin de dañar la integridad de Wikipedia. A estas ediciones se les denomina Vandalismo

[1] http://www.alexa.com/topsites

y su detección y reversión es objeto de estudió; sin embargo, por su crecimiento y aumento de ediciones es una tarea bastante compleja que se necesita del uso de sistemas automáticos que detecten el vandalismo mas rápidamente que un ser humano para minimizar el daño realizado.[8]. Por otro lado, se estudia el impacto de las ediciones con propósitos vandálicos afectando a los lectores de la Wikipedia calculando el número de visitas o consultas de cada artículo en el tiempo que se mantiene en este estado [6]. El presente trabajo tiene el objetivo calcular el impacto que existe por el vandalismo en la Wikipedia en un mes con el objetivo de comprender si el daño provocado es de relevancia respecto al número de consultas que recibe.

Wikipedia tiene un número de artículos muy extenso sobre su estudio iniciando por el estudio de su crecimiento inicial, como llego a tener el tamaño que tiene, estudios sobre la detección del vandalismo ([8], [2], [10], [6]), evaluación de la reputación de un usuario y Comprobación de la calidad de los artículos incluso hay página web dedicadas a recopilar todos los estudios sobre ella como Wikilit, Wikipaper, AcaWiki y la misma Wikimedia promueve su investigación [7] .

La analítica visual se ha usado en muchas investigaciones de la Wikipedia destacando el trabajo de Vi'egas Fernanda en el 2004 que a raíz de su trabajo se plantearon muchos estudios posteriores sobre la colaboración de los editores dentro de un artículo o de como el vandalismo afecto a los artículos [10] y el trabajo de Taha Yasseri que utiliza grafos para representar la colaboración entre editores de un mismo artículo [11]. El presente trabajo establece trabajos similares en la analítica visual como en el cálculo de visitas y el impacto de vandalismo en Wikipedia discutiendo como difiere de los resultados de las anteriores investigaciones. En la siguiente sección se describe los conceptos básicos de la Wikipedia y como se establece y mide su daño. Y por último se explica como se calculó el impacto del vandalismo de forma horizontal (por artículo) y de forma vertical (por hora del día o mes) en específico y se discutirá como estos valores son de interés para comprender el funcionamiento de Wikipedia. Este trabajo ha sido realizado con el motivo de entender como funciona una plataforma colaborativa dentro de nuestra sociedad y como en la práctica llega a ser más que un sitio web de referencia para convertirse en un centro de discusión sobre el conocimiento humano.

2. Objetivos

El objetivo es analizar el vandalismo en una ventana de tiempo identificando su duración, los autores y su efecto en las consultas o visualizaciones realizadas a los artículos de Wikipedia en una ventana de tiempo y como en su totalidad afectan la integridad de la plataforma.

- Aplicar algoritmos que identifiquen en un número de revisiones que existió vandalismo utilizando las propiedades que posee una revisión como el autor, el comentario de la revisión, grupo del usuario. hash de la revisión, etc.

- Extraer de forma eficiente el número de visitas de un artículo de los archivos de registro de las consultas a Wikipedia.
- Desarrollar una herramienta para el análisis del efecto del vandalismo de un artículo.

3. Trabajos Relacionados

En el 2004 Vi'egas propuso una herramienta exploratoria para visualizar las revisiones de los artículos con el objetivo de comprender como funciona la colaboración dentro de un artículo. Vi'egas expone la herramienta de visualización actual que posee Wikipedia para la comparación de textos entre revisiones mostrando que es una herramienta "diff"(meld, sourceTree, etc.) que posee dos problemas: el primero solo muestra dos versiones al mismo tiempo y el segundo que solo ve la diferencia entre párrafos habiendo la probabilidad de confundir la naturaleza del cambio, un carácter de más podría identificarse como que el párrafo está eliminado, por lo tanto, propone una nueva técnica llamada "history_flow"que tiene el objetivo de hacer visibles inmediatamente las tendencias generales de los historiales de revisión preservando los detalles para una revisión posterior. Gracias a la propuesta se detectaron ciertas tendencias de colaboración en los artículos ayudando a comprender como funciona la Wikipedia. Se detectó el daño provocado por el limpiado completo de artículos y se detectó que el tiempo que dura en este estado es mínimo, un promedio de 2.8 minutos, se identificó 5 acciones de vandalismo más comunes: eliminación masiva, inserción de vulgaridades, inserción de texto sin contexto, inserción de enlaces anómalos, inserción de opinión eliminando el punto neutral de los artículos. Otra tendencia identificada es *zigzag* identificando las denominadas *guerras de ediciones* que son descritas como la lucha de opiniones entre dos editores que desean publicar su versión del artículo este son de fácil detección gracias a la técnica de history flow observando el patrón zigzag. El presente trabajo propone una técnica de visualización que se enfoca en la duración del vandalismo y su efecto, utilizando la metadata de las revisiones en series de tiempo identificando la duración del vandalismo si lo hubiera. (Ver figura. 1). Otro ejemplo de visualización que muestra el comportamiento de la colaboración en un artículo es el ejemplo de Taha Yasseri [11] que muestra por medio de un grafo de las revisiones como los editores colaboran, donde el enlace es entre editores sucesivos por lo que podemos ver si un editor recibe muchas revisiones por el tamaño del diámetro y por el ancho del enlace revisiones sucesivas entre autores. (Ver figura 2).

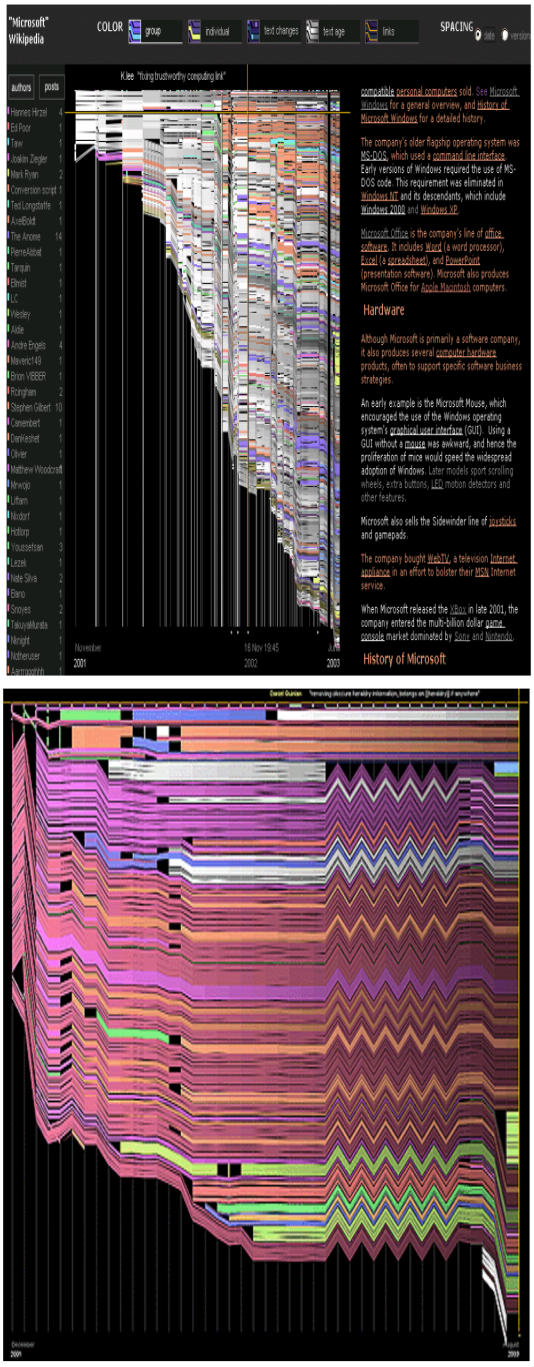


Figura 1: Técnica propuesta por Fernanda Viegas para la visualización de los artículos, visualizando el efecto zigzag de una guerra de ediciones.

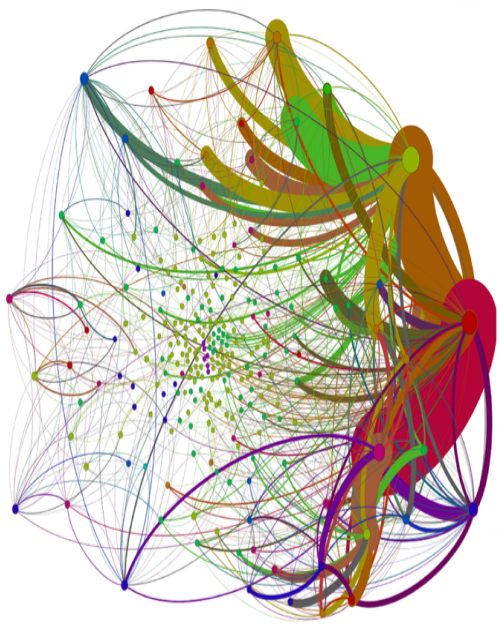


Figura 2: Grafo de editores del artículo *Anarchism* de la Wikipedia en inglés. Los nodos son editores y los enlaces son revisiones. El tamaño de los nodos es proporcional al número de enlaces y el ancho de los enlaces es dado por el número de revisiones entre usuarios.

Muchas plataformas de colaboración generan valor por su contenido por medio del filtrado y resolución de opiniones (slashdot.org, reddit.com, digg.com), pero la más ejemplar es Wikipedia con 1,218,643 artículos para el idioma español y en aumento (7 % para el Mayo 2016). El contenido se genera con una relación de poder en donde un número muy pequeño de editores genera el mayor número de ediciones. En la Wikipedia en español se ha comprobado que solo 28 usuarios (25 administrados y 3 usuarios registrados) han creado el 25 % de los artículos [7]. Podemos concluir que existe un gran número de usuarios que realizan pocas ediciones a la Wikipedia y que hay un grupo pequeño de editores que realizan el mayor número de ediciones (relación de poder).

Reid Priedhorsky en el 2007 extendió el trabajo de Vi'egas profundizando en detectar y categorizar más tipos de daños por vandalismo para su medición posterior observando cuanto tiempo están en estado vandálico y cuantas visitas o consultas hubo en ese tiempo. Reid propuso una forma de medir las visualizaciones proponiendo una tasa de visualización. Esta tasa se describe como el impacto del vandalismo sirviendo como base para el cálculo del daño proponiendo identificar todas las reversiones de un artículo por medio del hash del contenido de la revisión. Priedhorsky concluye que el vandalismo esta en aumento pero el daño provocado es mínimo a comparación del número total de visitas a la Wikipedia por lo que plantea la discusión si es necesario preocuparse por ciertos tipos de vandalismos a pesar de que Wikipedia clasifica cualquier intento de vandalismo como perjudicial. Esta propuesta se utilizara en este trabajo para la detección de las reversiones por ofrecer una manera muy simple y eficiente de detectar cuando ha ocurrido el vandalismo sin saber el contenido del artículo.

Los editores contrarrestan el vandalismo creando una nueva edición con el contenido de una edición antigua, Esta acción (volver a una edición previa) es lo que se denomina reversiónz es la recomendada por la propia Wikipedia para contrarrestar los actos de vandalismo. Su detección se realiza por lectores humanos que sugieren por medio de la página de discusión que el artículo puede haber sido afectado. También set utilizan programas automáticos que reducen la exposición este estado disminuyendo la carga de trabajo de los editores; Sin embargo, por el crecimiento de la Wikipedia el vandalismo también ha aumentado y evolucionado afectando el trabajo de su detección aun más difícil. Este vandalismo se le denomina "Vandalismo Sigilosoz se observa de muchas formas una de ellas es la reparación de contenido por vandalismo insertando contenido en estado vandálico, Khoi Nguyen propone la detección de este tipo de vandalismo por medio del reconocimiento del contexto de las palabras de un artículo, como resultado se creó un clasificador que clasifica de forma efectiva (80 % de las veces) las oraciones detectando anomalías como palabras ofensivas o palabras fuera del contexto del artículo. [9]. En este trabajo no se explora como detectar el vandalismo que está ocurriendo en el momento ya que no es parte de la investigación, nos centraremos en la medición del vandalismo que afectó a un artículo y de donde proviene el daño.

En el 2012 Mirko Kampf [5] analizó de forma extensiva el comportamiento de los accesos a los artículos detectando ráfagas o picos de consultas y la relación

con el tiempo en que se realizan las ediciones. Concluye que las propiedades estadísticas que presentan las ediciones son más cercanas a ruido blanco.[es] decir que por la complejidad que presenta el modelo de ediciones que utiliza Wikipedia a largo plazo destruyen cualquier correlación que pueda existir entre las ediciones. Este trabajo explorara como ver esta información de forma que se vea en una serie de tiempo la tendencia de las ediciones y las consultas.

4. Wikipedia

Wikipedia se define a sí misma como una *enciclopedia libre políglota que cualquier persona puede editar cualquier contenido.* Actualmente la administra la Fundación Wikimedia sin ánimo de lucro que es mantenida por donaciones de los mismos lectores. Se creó a partir del año 2001 por Jimmy Wales y Larry Sanger como producto de otra enciclopedia libre llamada Nupedia que sirvió como proyecto padre; Nupedia utilizaba un proceso más tradicional para la creación de contenido donde los artículos eran escritos por expertos y evaluados por un proceso de edición riguroso; Sin embargo, solo creó 25 artículos para finales del primer año. utilizaba el concepto de Wiki inventado por Ward Cunningham en el año 1995 definidas como páginas en donde cualquier lector también puede ser autor ([10] ofreciendo herramientas de edición,[1]). Wikipedia empezó a tener vida propia creando 2000 artículos para el primer año que a comparación de Nupedia los usuarios podían editar los artículos sin la aprobación de los administradores.

Wikipedia se compone de más 280 ediciones, una por idioma. Los conceptos y estructuras en los que se basan son muy similares entre las ediciones variando en modificaciones locales de la comunidad de editores por edición. La estructura está conformada por dos elementos esenciales: los artículos y los editores que se hacen llamar "wikipedistas", el resto de elementos ayudan a realizar conexión entre artículos y editores o entre los mismos editores. El contenido de Wikipedia también se crea fuera de la página de los artículos usando las páginas de discusión, de comunicación, o discusión general [11].

4.1. Elementos de la Wikipedia

- **Artículo** Consiste en entradas de diferentes temas que en esencia posee un título, un contenido y un historial que tiene todas las revisiones previas que ha tenido el artículo. Cada artículo está conectado a otros artículos por medio de enlaces web convirtiendo la Wikipedia en un inmenso grafo. En general el artículo puede ser editado por cualquier usuario, pero hay artículos que están protegidos contra edición y mecanismos que protegen el contenido contra vandalismo. Entre los artículos hay artículos que son considerados los mejores de Wikipedia evaluados por los mismos editores y artículos controversia les que poseen en su historial evidencia de guerra de ediciones. Los artículos nuevos creados por usuarios que cuentan con el permiso de verificado o auto-verificado aparecerán, por defecto, como verificados. Ello permite a los usuarios verificadores centrarse en aquellos artículos donde el riesgo de

violación de las políticas es mayor [7]. Al momento se ha modificado aparecen en la página especial de *CambiosRecientes*. Cualquier usuario de Wikipedia puede patrullar los cambios recientes para detectar ediciones dañinas, esto es, ediciones que no siguen las reglas de la Wikipedia sobre el contenido, a menudo errores sin malicia de usuarios nuevos.

- **Wikipedistas** En esencia cualquier persona con acceso a internet puede ser editor de Wikipedia. Si lo desea puede editar anónimamente donde la plataforma usará la IP origen como forma de identificación del usuario o crear un perfil con información del usuario qué Wikipedia no revelara si no se desea. También se guarda de donde son los editores, pero esta información no es del todo precisa dificultando el análisis del origen de las colaboraciones a la plataforma. Existen 4 niveles de acceso:
  - **Usuario Administrador** con el permiso de editar páginas protegidas, eliminar o proteger páginas y bloquear a los editores.
  - **Usuario Anónimo** con el permiso de editar artículos no protegidos
  - **Usuario Confirmado** con el permiso de editar artículos semiprotegidos y renombrar artículos.
  - **Nuevo Usuario** con el permiso de editar artículos no protegidos y crea nuevos.
  - **Burócratas** que poseen el permiso de editar los permisos de los demás Wikipedistas.
- **Políticas y Guías** Son páginas especiales que documentan las buenas prácticas aceptadas por la comunidad de Wikipedia. Hay 4 enunciados en los que se basa las leyes que rigen la Wikipedia en español.
  - **Wikipedia es una enciclopedia** y por consecuencia no es un periódico tampoco una plataforma de propaganda ni un diccionario y por lo tanto no se debe considerar como una fuente primaria.
  - **Wikipedia busca siempre el punto neutral**. Ofrecer la información en todos los puntos de vista posibles sin presentar un punto de vista como el mejor.
  - **Wikipedia y su contenido es libre** bajo la licencia Creative Commons.
  - **Wikipedia sigue unas normas de etiqueta**. Respeta a tus compañeros incluso al no estar de acuerdo con ellos evitando ataques personales y generalizaciones. Ser abierto, acogedor e inclusivo.
  - **Toda Ley excepto las cuatro anteriores no es permanente** por lo que se motiva a editar y agregar nuevo contenido a la Wikipedia.

  Establecen las bases de la colaboración entre los editores e incentivan a que cualquier usuario pueda editar porque Wikipedia registra todo contenido, si es necesario, se puede recuperar en cualquier momento; Sin embargo, por su naturaleza abierta y colaborativa provoca que existan ciertos problemas. El cofundador de Wikipedia afirma recibir 10 correos electrónicos cada día de estudiantes que han reprobado algún curso gracias a los datos erróneos de la Wikipedia [4]. Estos datos erróneos son producto de ediciones en vandalismo.
- **Páginas de Usuarios** Se usan para proveer información personal del editor y de contenido relacionado. se rige por las mismas reglas que los artículos.

- **Páginas de Discusión** Provee un espacio de discusión sobre los cambios de un artículo o proyecto. Es el sitio principal para resolver desacuerdos y conflictos de edición
- **Páginas de Comunicación** Diseñados para una comunicación directa general entre editores. Usualmente para conversaciones más personales.
- **Páginas de Discusión generales** Aparte de las páginas de discusión de un artículo y las páginas de comunicación hay páginas para actividades colectivas.
- **Categorías** Agrupan diferentes artículos sobre una misma temática.
- **Programa Automáticos** Son programas que ayudan a mantener la integridad de la Wikipedia modificando los artículos. Normalmente su función su función consiste en la reparación ortográfica y gramática de artículos como de detección de vandalismo aunque su uso se ha extendido en ser utilizado para la migración de páginas y otros procesos de mantenimiento de datos.

El Wiki se basa en una tecnología que permite a los visitantes crear actualizaciones instantáneas a una página web por medio de una interfaz de edición. Cada página de la wiki de un artículo tiene el enlace *editar* donde cualquier lector puede editar el contenido y remplaza el contenido actual. Editar hace uso de marcadores que son traducidos a código HTML. Cada usuario de una Wiki puede registrarse o usar la plataforma como usuario anónimo o registrado donde son reconocidos por la IP. Todas las Wikis poseen un sistema que graba todas las ediciones de un artículo permitiendo facilitar la reversión a una versión anterior de forma fácil. Este sistema asegura que no exista un daño permanente a causa de una mala edición. Un artículo en wikipedia se presenta en una pagina web que esta compuesta de 4 secciones principales: el contenido del artículo, discusiones sobre el contenido, historial de ediciones y la sección para editar el contenido propias de un página wiki (ver figura 3). La sección de historial de ediciones presenta toda las ediciones que se han creado para el presente artículo mostrando una herramienta para comparar cualquier edición con otra edición, también podemos observar la diferencia de caracteres entre ediciones y el estado de la edición (si ha sido eliminado o esta desactivada).

4.2. Estadísticas de la Wikipedia en Español

La Wikipedia en español es la cuarta por el número de páginas, es la décima por el número de artículos enciclopédicos y es la segunda en número de visitas por hora. [7] Para el mes de Mayo 2016 existen 1,218,643 artículos en la Wikipedia [1] en español teniendo un crecimiento del 7 % anual; un promedio de 237 artículos nuevos por día. En promedio hay 493,549 ediciones en el mes de mayo. En la figura 4 se observa una disminución del acceso a Wikipedia por la página principal y hay un incremento en su acceso por plataformas móviles y que la cantidad de accesos es dependiendo del mes y los eventos del mundo, en la figura 5 muestra que la cantidad de ediciones es mayor a las reversiones lo que

[1] `https://stats.wikimedia.org/EN/SummaryES.htm`

Figura 3: Secciones de un artículo de Wikipedia / Historial de un artículo de Wikipedia

podemos concluir que la mayoría de contribuciones a la Wikipedia perdura en el tiempo. En el 2015 Ángel Zazo Rodríguez refuerza esta idea identificando que el 67 % de artículos no se revertido nunca y el 13 % lo ha sido solamente una vez, por lo que se puede afirmar que hay una buena disposición por parte de la comunidad editora que el contenido perdure.

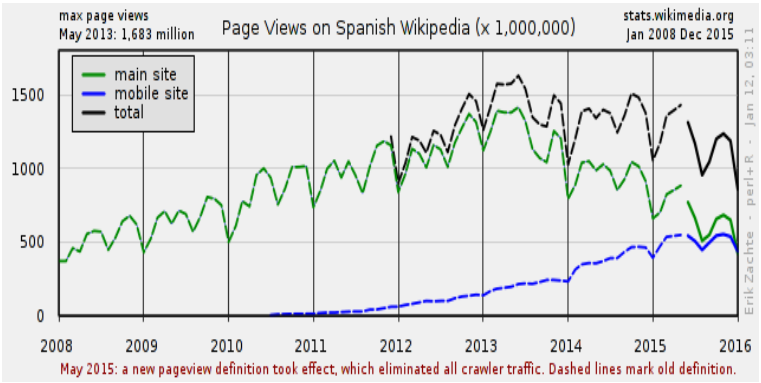


Figura 4: Gráfica del número de visitas en desde el año 2008 al 2016. (Wikimedia edición idioma español Mayo 2016)

Se analizó el año 2014 (primero de enero a las 00 horas a primero de enero de 2015 a las 00 horas) . Se crearon 5,625,359 ediciones, 76,853 nuevos artículos y empezaron a editar 669,216 nuevos usuarios. Para inicios del 2015 el 43 % de artículos han sido revertidos al menos una vez. Club_América, Dragon_Ball y Venezuela son las páginas más revertidas (Ver tabla 1).

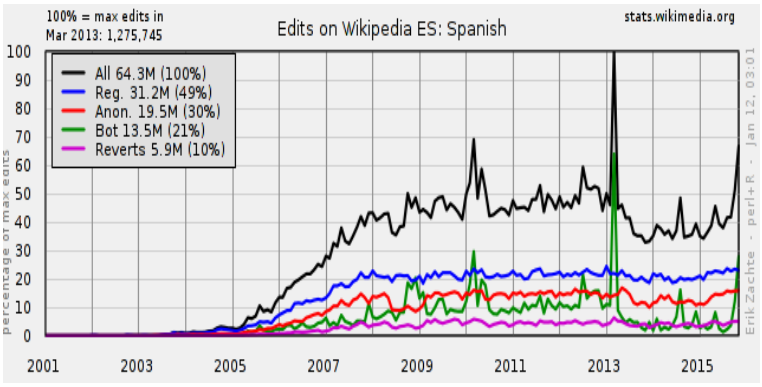


Figura 5: Gráfica del número de ediciones del el año 2008 al 2015 (Wikimedia edición idioma español Mayo 2016)

La calidad de la Wikipedia ha sido estudiada para confirmar si el contenido cumple con el de una enciclopedia, resulta ser que su contenido es de una calidad similar[6]. Sin embargo, por el vandalismo que puede existir en los artículos no es recomendable usarla como referencia de información. Harvard establece que el contenido de Wikipedia no debe ser utilizado como referencia para artículos científicos si no utilizarlo como fuente básica para empezar a comprender un tema y utilizar otras fuentes verificadas al profundizar. [2].

Tabla 1: Artículos más Revertidos

Título	Número de Reversiones
Club_América	2081
Dragon_Ball	1960
Venezuela	1951
Baloncesto	1930
Club_Atlético_Peñarol	1849

Hay 3.6 millones de usuarios registrados que en la práctica solamente han editado artículos 557,970. El número de bots es elevado si es comparado con el número de administradores; resulta que la mayoría de programas automáticos o bots están activos 1 o dos días y hay un grupo pequeño de robots que esta siempre activo que realizan miles de ediciones (31 bots han realizado 7.2 millones de ediciones) [7].

[2] `http://isites.harvard.edu/icb/icb.do?keyword=k70847&pageid=icb.page346376`

4.3. Daño de la Wikipedia

Wikipedia define el vandalismo como *Cualquier adición , eliminación o cambio de contenido hecho de forma deliberada que compromete la integridad de Wikipedia* [3]. El vandalismo puede tomar muchas formas, el eliminado completo del contenido de un artículo , modificar el contenido para que sea difícil cargar el artículo, agregar contenido ofensivo, no relacionado, sin sentido, información inexacta, spam, etc;

Tipos de vandalismo:

- **Basura** Agregar contenido sin fundamento o sin ninguna referencia para ser verificada
- **Grafiti**: Agregar contenido fuera del contexto del documento
- **Insultos** Agregar palabras que demigran a alguien o algún punto de vista provocando que el artículo no tenga un punto neutral
- **Chistes** Agregar contenido con el fin de hacer burla del mismo.
- **Blanqueo de Paginas** El limpiado completo de páginas o borrado de contenido.
- **Vandalismo en páginas de usuarios**: El vandalismo también se da en la edición de la página de los usuarios con el de hacer chiste o demigrar al usuario.

De los artículos mas reconocidos por el impacto del vandalismo que ha sufrido podemos destacar:

- En el 2016 un joven de 12 años altero el artículo de la lista de primer ministros de Australia escribiendo su nombre como el actual primer ministro y cambiando su página de perfil para reflejar el cambio (fuentes Huffington Post, Mashable abril 2016). [3]
- En el 2010 el el New York Times publicó que el juez Roger Vinson que anunció permitir un desafío legal a la nueva ley de salud para avanzar a una audiencia completa fue atacado en su artículo bibliográfico marcándolo como una persona que caza animales y que practica la taxidermia provocando que el grupo de personas que lo apoyaban no le dieran su voto en la audiencia. El juez Roger Vinson es presidente *American Camellia Society* que promueve el cultivo de plantas del genero Camellia (fuente The New York Times) [4].
- En el 2005 el ministro de noruega Jens Stoltenberg encontró su artículo bibliográfico con varias oraciones difamatorias y Adam Curry altero la pagina de Podcast para remover el crédito a otras personas que contribuyeron en la creación del mismo . [6].

Reid Priedhorsky categoriza 7 tipos de vandalismo identificando que el mas recurrente es la inserción de Spam o contenido sin relación con el artículo (53 % del vandalismo total en la Wikipedia) y que de segundo con un 28 % es la inserción de contenido ofensivo (Ver tabla 2). Su medición de su impacto en los artículos es de suma importancia porque la calidad de una edición es utilizada para evaluar la contribución de un editor.

[3] `http://mashable.com/2016/04/13/wikipedia-australia-prime-minister/`
[4] `http://www.nytimes.com/2010/09/16/us/16judge.html?_r=0`

Tabla 2: Tipos de vandalismo mas común (fuente Reid Priedhorsky 2007) [6]

Tipo	%	Descripción
Ediciones sin Sentido al contexto del artículo	**53**	Insertar contenido que no pertenece
Ediciones ofensivas palabras ofensivas	**28**	Insertar contenido que contenga
Desinformación	**20**	Insertar información que no es verdadera
Borrado parcial	**14**	Se borra parcialmente contenido de artículos
Spam o links a otros artículos	**9**	Insertar propaganda por medio de palabras
Borrado total	**9**	Borrado total del contenido de un artículo
Otros en los anteriores	**5**	Cualquier tipo de vandalismo que no esta

Los usuarios con atributos de reversor (administradores y algunos bots) son aquéllos que expresamente patrullan páginas para revertir vandalismos. Muchos usuarios registrados, sobre todo los más activos, también dedican mucho esfuerzo en esta labor. El número total de ediciones vandálicas revertidas por todos estos usuarios ha sido de 3,5 millones; es decir, que al menos otro número igual o mayor ha sido fruto del vandalismo [7].

4.4. Otros efectos detectados

- **Fusión de páginas** En la Wikipedia existen procesos para la unificación de dos páginas individuales dentro de una para crear un contenido unificado, este proceso para efectivo de unir los dos historiales de revisiones de las dos páginas dentro de una y esto provoca ciertas anomalías en la forma que se revierten los datos.
- **Movimiento de páginas** Cuando se crea una nueva página con el contenido de otra. Se da cuando se desea mover el contenido de un artículo a un título mas adecuado.
- **Guerra de Ediciones** La guerra de ediciones es un fenómeno que se identifica igual que una reversión por vandalismo pero es un fenómeno de las plataformas de edición abierta donde los editores se revierten constantemente entre si.
- **Ediciones sucesivas** Un editor edita sucesivamente la misma página creando revisiones sucesiva porque desea observar el resultado de su edición.

4.5. Datos de la Wikipedia

Wikipedia funciona gracias al software libre MediaWiki. Todas las acciones que se realizan en Wikipedia quedan registradas en bases de datos. Un aspecto interesante desde el punto de vista de la investigación, es que la Fundación Wikimedia mantiene el historial de todas las acciones que se realizan en sus

proyectos, y este historial, a excepción de los datos personales de los usuarios registrados, es público [7].

Se pueden acceder de 2 formas los datos de la Wikipedia, en tiempo real utilizando la interfaz de MediaWiki MediaWiki tool labs (urlttp://wikitech.wikimedia.org)) o si se desea los datos estadísiticos hay páginas como Wiki Stats que ofrece las estadísticas mas actualizadas de la Wikipedia (`http://stats.wikimedia.org/`). La Segunda forma es accediendo al volcado de datos de los proyectos de MediaWiki que almacena toda la información de forma regular (`http://dumps.wikimedia.org`). Generalmente esta fuente de datos es muy útil para el análisis de artículos ya que la fuente más confiable son estas copias estáticas [11].

5. Metodología

Para cumplir con los objetivos de la investigación se realizaron dos tareas de Pre-procesamiento de datos, la primera era obtener de forma eficiente las consultas de la Wikipedia en español y el segundo identificar solo con el historial de revisiones cuando ha ocurrido vandalismo. Toda esta información fue guardada en una base de datos para su posterior consulta por medio una herramienta de análisis. Los datos ya guardados se usaron para realizar representaciones de visuales que explican las tendencias de los datos.

5.1. Pre-Procesamiento

Los datos que se analizaron para este trabajo pertenecen al volcado de Wikipedia del 05-01-2015 que posee la información desde que se creo la Wikipedia hasta inicios del año 2015 y los datos de visualización o de consulta de artículos desde 2009 a 2014, datos por hora. También su utilizo el servicio *Wikipedia article traffic statistics* para el análisis de los accesos que sirve como fuente para la herramienta de análisis desarrollada.

Extracción de los accesos Se descargaron los registros de la visualización de los años 2009 a 2014 de la página de volcados de datos de la url `https://dumps.wikimedia.org/other/pagecounts-raw/` que posee los datos del 2007 al mayo del 2016; cada año agrupa directorios por mes y cada mes guarda los archivos comprimidos de los accesos en formato *.gz*. Estos datos fueron previamente recopilados por el grupo de investigación y filtrados para el dominio *es* que pertenece al idioma español por lo que el archivo solo contiene las consultas para esta edición. El archivo original comprimido tiene un aproximado de 40 megabytes de tamaño con 5 millones de registros. Al filtrarlos cada archivo tiene un tamaño aproximado de 3.5 megabytes comprimidos y 22 megabytes descomprimidos y un aproximado de 270,000 lineas para cada hora del día. Si procesáramos todas las consultas necesitaríamos un máximo de 25 gigas de espacio para almacenar todo el año 2014 solo para la edición en español.

La recopilación de la información de las consultas por artículo en Wikipedia se realiza extrayendo la información de archivos comprimidos en formato *.gz*

que fueron creados de los archivos originales de registro de las consultas a las páginas. Cada archivo comprimido posee una linea con la información de a que dominio pertenece la petición, páginas solicitada, cuantas veces se ha ejecutado esa petición a lo largo de 1 hora y el número de bytes transmitido (Ver tabla 3).

Tabla 3: Estructura de la información de una petición a un página de Wikipedia

Dominio	Página	Número de peticiones por hora	Cantidad de bytes transmitidos
es.d	Andorra	346	2345

Al momento de extraer la información nos encontramos con 4 problemas. El primero es poder extraer la petición de una consulta de forma que pueda igualarse al título guardado en la base de datos porque el registro no posee ningún otro dato para enlazar la información de una página a un registro del archivo de peticiones por lo que se debe transformar la información de forma que se pueda igualar. Para consultas sencillas como puede ser *Venezuela*, *Madrid* no hay problema al comparar con la información de la base de datos, pero, al ver consultas con tildes y con eñes por ser la Wikipedia en español hay consultas como *Club_Am%C3%A9rica, Peri%C3%B3dico, Ni%C3%B1o* que pertenecen a Club_América Periódico Niño respectivamente que se deben transformar para ser comparados. Por lo anterior se realizo 2 tareas de transformación. La primera es convertir los código especiales de escapeado de urls a sus caracteres respectivos con el fin de ya no tener una url como consulta si no palabras del idioma español, este proceso se decidió que debía ejecutar al menos dos veces por consulta porque se detecto que había url dentro de la url y esas no existían en la wikipedia como página de consulta por lo que se volvió a procesar esta para ser comparadas. Segundo como son dos fuentes de datos, una base de datos y archivos de registro comprimidos, la codificación de los repositorios es distinta por lo que es muy difícil comparar cadenas de caracteres especiales entre diferentes codificaciones (como la ñ). Se descubrió que la mejor forma es convertir toda cadena de caracteres a una representación de bytes porque sin importar la codificación tiene los mismos número de bytes. Se utilizó la representación hexadecimal de las dos cadenas de caracteres, de la base de datos utilizando la función HEX, y de los archivos de registro utilizando el paquete binascii.hexlify, se compararon como números. Estos dos procesos facilitaron poder comparar y extraer de forma iterativa por cada archivo el número de visitas por petición.

El tercer problema se da por la forma en que funcionan las redirecciones en la Wikipedia porque una persona puede escribir como petición *Espana* y esta redirige a la página *España* por lo que hay que unificar toda las peticiones de todas las redirecciones para una página, este proceso es laborioso porque hay que obtener por cada petición su lista de redirecciones o identificar que es una redirección y sumar las visualización a la página destino. Este proceso es muy laborioso, por lo que primero se identifico las redirecciones registradas por cada página de la tabla original de la base de datos de Wikipedia y se unificaron en

una nueva tabla *pageOK* que posee solo las páginas destino de toda re-dirección (se redujo de 2.7 millones de páginas a 1,1 millones). El cuarto problema fue al momento de descargar los archivos de registro de peticiones. El nombre del archivo comprimido esta dado por el siguiente formato "20140101-000000"(año, mes, día - hora, minuto, segundo) pero para algunos archivos el nombre no termina en el segundo 00 por lo que puede terminar en cualquier segundo del primer minuto 00-59 por lo que para descargarlos se realiza la petición pero si no logra descargar el archivo se realiza la petición con un segundo mas 1.

petición (1)	página (2)	hexadecimal (3)
Club_Am%C3%A9rica	Club_América	436c75625f416de972696361

Tabla 4: Transformación de la información de una petición

5.2. Base de Datos

La base de datos esta compuesta de 5 tablas extraídas de las tablas originales de Wikipedia. Esta base de datos posee la información de todas las revisiones y páginas creadas hasta inicios del año 2015 teniendo un total de 55,473,131 revisiones con un total de 6,455,206 editores y 1,111,800 páginas, Esta información fue extraída con anterioridad del volcado de datos 2015-05-01 de las tablas de la Wikipedia. Cada tabla es un extracto de una o mas tablas de la estructura original.

Se utilizaron las siguientes tablas:

- la tabla **revOK** almacena el historial de una página agregándole información perteneciente a la tabla usuario como es el grupo al que pertenece (si es anónimo, robot, administrador, o un usuario registrado) y el texto del nombre del usuario. La tabla es una union entre la tabla *user* y la tabla *revision* de la estructura original de base de datos extrayendo de la tabla *user* el grupo del usuario que posteriormente servira para la detección del vandalismo y el nombre del usuario y de *revision* todos los campos (Ver tabla. 5).

Se crearon dos tablas en la base de datos para almacenar el acceso por hora o por día de un artículo y para almacenar el detalle de las reversiones por vandalismo con el fin de ser utilizados como cache de información para no tener que volver a procesar la data. El llenado de estas tablas se detallara en la siguiente sección.

- **Consumo** posee la información de los accesos por hora y por día de un artículo.
- **VandalismOK** posee la información de todos las revisiones de vandalismo, entre sus campos esta la duración del vandalismo, el editor que inicio el acto vandálico y la persona que lo reparo.

Campo	Descripción
rev_id	id de la revisión o edición
rev_page	id de la página
rev_user_group	grupo del usuario (administrador, anónimo, programa automático, usuario registrado)
rev_timestamp	fecha y hora de la revisión
rev_sha1	hash generado del contenido de la revisión
rev_parent_id	id de la revisión anterior según el historial del artículo

Tabla 5: tabla revOK campos principales

Detección del Vandalismo Cuando se realiza una edición en Wikipedia se guarda el contenido anterior en su base de datos, es decir, que tiene almacenado todo el historial de los cambios de un artículo. Cada edición tiene un hash (rev_sha1) único que lo representa. Cuando sucede una reversión (se recupera la información de un artículo por una versión antigua) se mantiene el hash de la versión. En el ejemplo de la tabla 6 la revisión 708 es una reversión porque posee el mismo hash (sto2oy) que la revisión 679. Gracias a este comportamiento de las reversiones nos ayudan a detectar posible vandalismo, sin embargo, no es suficiente porque no todo usuario se considera un reversor, los únicos que son considerados son los administradores y los bots o programas automáticos. En el ejemplo la revisión 708 se considera una reversión por vandalismo por ser de un administrador [6].

id revisión	usuario	grupo	sha1	fecha edición
617	PatruBOT	Bot	jk0vkl	2014-07-11 23:30:21
665	Chico512	U. Registrado	eaogy	2014-07-17 02:42:42
679	Syum90	U. Registrado	sto2oy	2014-07-23 09:50:30
705	UA31	U. Registrado	g5a5m	2014-07-30 21:58:49
706	UA31	U. Registrado	2l2bkt	2014-07-30 22:10:49
708	Gusama Romero	Administrador	sto2oy	2014-07-31 01:29:02
709	PatruBOT	Bot	07syy	2014-08-01 10:54:00

Tabla 6: Ejemplo de una reversión en la tabla revOK

5.3. Medición del Vandalismo

Partiendo del conocimiento de poder identificar cuando sucede una reversión por vandalismo se obtiene la edición que posee como revisión padre la penúltima

edición con el sha1 que posee la edición resultante de la reversión, es decir, se obtiene la primera revisión que es considerada contenido erróneo para el artículo. En la tabla 7 se ilustra este comportamiento de las reversiones identificando con colores los diferentes tipos de edición. Se utiliza el tiempo en que se registra la primera edición en vandalismo y se resta del tiempo de la revisión resultante de la reversión porque a partir de esta edición se elimino el contenido vandálico.

Tabla 7: Tabla de tipos de revisiones

id	rev1	rev2	rev3	rev4	rev5
sha1	ser2	23jm	2i3om	ser2	rios

- **verde** revisión antigua a la que ser revertió el contenido
- **rojo** revisión que se considera vandalismo
- **naranja** revisión resultante de revertir a la revisión origen (verde)
- **azul** revisión que persiste por agregar valor al artículo

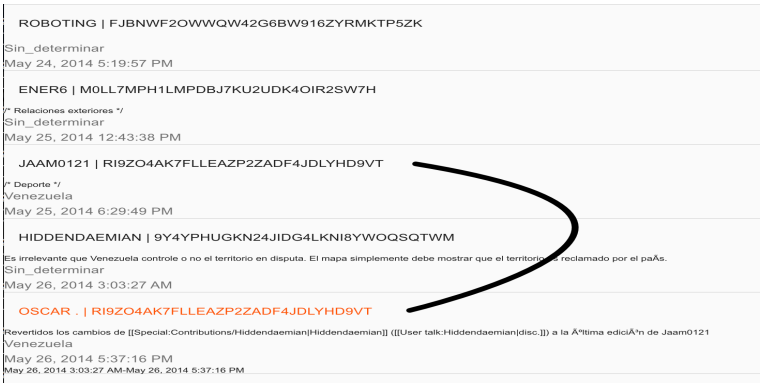


Figura 6: Ventana de revisiones ordenadas por fecha de un artículo de la herramienta de análisis que muestra la reversión de una edición

Medición de las Visitas en Vandalismo Para el calculo del número de visualizaciones que han sido realizadas cuando el contenido se encuentra en estado vandálico, se realiza por medio de un aproximado del porcentaje de cuanto vandalismo ha afectado en un plazo de tiempo (Ec. 1). Para comprender mejor este concepto veamos la tabla 8 donde cada columna es una hora del día y una revisión de vandalismo puede durar una fracción (ej:el 30 %) de ese tiempo por lo que se cálculo el porcentaje de tiempo dentro de la hora del día al que pertenece; Sin embargo hay reversiones que se ubican entre dos horas o más por lo que

hay que identificar a que horas afecta y que porcentaje del fragmento de tiempo pertenece al vandalismo (ej:las celdas en amarillo). Ya calculado el porcentaje se multiplica por el número de visitas del fragmento de tiempo (Ec. 2).

$$\frac{\#tiempo - en - vandalismo}{\#fragmento - de - tiempo} = \%en - vandalismo \quad (1)$$

$$\%-en-vandalism * \#visitas-en-tiem = \#visitas-en-vandalismo \quad (2)$$

Tabla 8: Distribución del efecto de una edición en plazo de 1 y 2 horas

hora del día	01	02	03	04	05	06
visitas	300	350	310	290	320	280
porcentaje del tiempo en vandalismo	10 %	10 %	0 %		30 %	
visitas afectadas	30	35			96	

- **amarillo** reversión que se encuentra entre dos horas
- **verde** reversión que se encuentra dentro de una sola hora

5.4. Herramienta para la Visualización del Efecto del Vandalismo

Para estudiar el efecto del vandalismo se decidió la creación de una aplicación que mostrara la información de las edición de un artículo en específico con el objetivo de comprobar el algoritmo de detección de vandalismo y el cálculo de las visualizaciones afectadas.

Implementación Se crearon 3 elementos, el analizador, el visualizador y el extractor de la información. Tanto el extractor como el analizador fueron escritos en el lenguaje python versión 3.5.2. El visualizador de la información esta escrito en electron (`http://electron.atom.io`) utilizando angularjs y d3 para la visualización y la interfaz gráfica. 7. Se utilizo esta arquitectura por ofrecer dos ventajas, la primera es el poder utilizar tecnologías web en un ambiente nativo y la segunda es poder utilizar la característica de multi-procesos para el procesamiento de varios artículos a la vez.

- **Analizador** Es un servidor web montado en Flask (`http://flask.pocoo.org`) que tiene la función de ir a la base de datos , realizando el mínimo número de consultas, para el análisis del vandalismo solo realiza dos consultas, y transforma a una representación json los datos. El servidor funciona por medio de la tecnología REST y tiene los siguiente servicios:

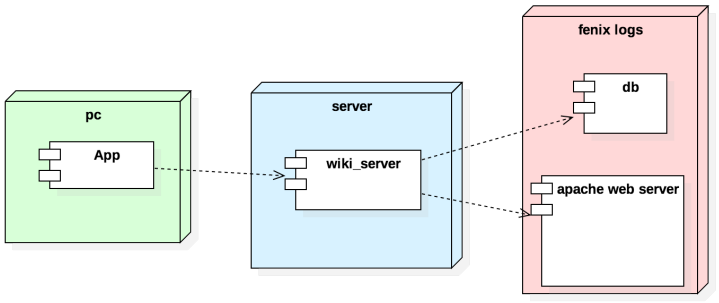


Figura 7: Distribución del sistema

- *analizar/vandalismo* recibe en json el id de la pagina, la fecha de inicio del análisis y fecha final. Itera sobre las revisiones que de la página en el rango de fechas identificando reversiones de posible vandalismo al identificar que la revisión es un usuario administrador o bot. Esta identificación es posible porque internamente el servicio guarda todo hash de revisión que itera por lo que si identifica una reversión , es porque ya guardo con anterioridad ese mismo hash. También el servicio itera sobre los accesos que hay por día entre el rango de fechas y al identificar que existe vandalismo en una hora se calcula el porcentaje que representa dentro de la hora. El servicio como resultado devuelve un json con la lista de todas las revisiones identificando si son reversiones de vandalismo y si lo son posee el detalle de la duración de la reversión, el número de visitas afectadas el usuario vándalo y el usuario reversor. Es importante resaltar que el análisis de la duración del vandalismo se facilito mucho gracias al uso del tipo de dato timedelta (`https://docs.python.org/3/library/datetime.html`) porque es una clase creada para el manejo de fragmentos de tiempo que facilita el calculo de horas, días, minutos, segundos, etc, entre dos fechas.
- *obtener/pagina* recibe en json el nombre de la página que se desea analizar y da como resultado el id al que pertenece la página.
- *obtener/consultas* recibe en json el id de la pagina, la fecha de inicio del análisis y fecha final y da como resultado una lista de todos los días ordenados con el número de consultas o visualizaciones.

Todos los servicios fueron configurados como servicios POST por la facilidad de mandar peticiones muy largas y con caracteres especiales, en cambio si hubieran sido GET habría que escapear el contenido antes de mandarlo al servidor.

- **Extractor** Son scripts escritos en python con el fin de extraer la información de las consultas para ser insertados a otro repositorio de datos. Se utilizaron en general dos scripts:
  - *extractor de visitas por hora* se probaron varias formas de extraer la información de forma rápida en tiempo real sin embargo la lectura de los

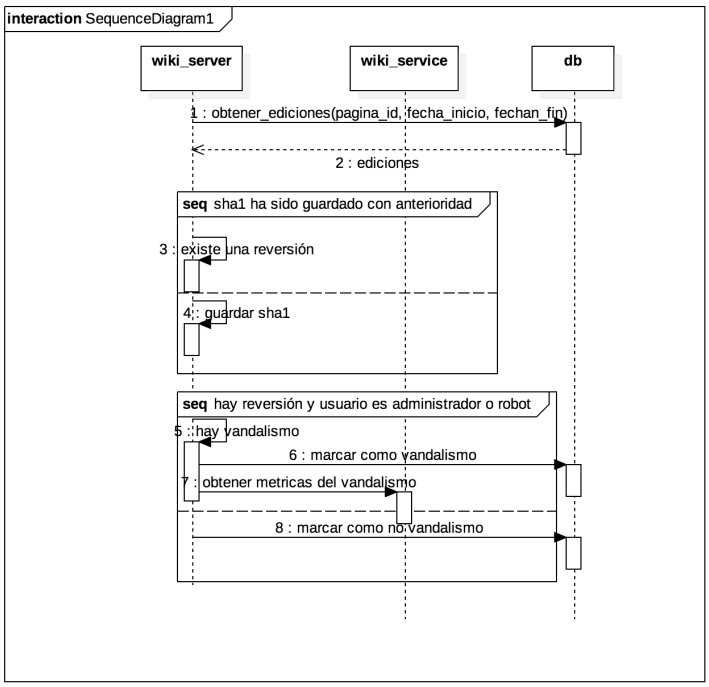


Figura 8: Proceso de detección de vandalismo

archivos resulto ser muy lenta por el número de filas a recorrer por archivo. Por lo que se decidió por realizar un proceso previo de extracción a un repositorio de datos y realizar las consultas al repositorio. La implementación mas rápida fue el guardar todas la páginas y sus ids en memoria en un diccionario donde la llave era la representación hexadecimal del titulo de la página y su valor el id iterando el archivo comprobando si la consulta en representación hexadecimal esta dentro del diccionario. Se utilizo la librería gzip para leer los archivos porque tiene la ventaja de administrar todos los archivos temporales sin necesidad de preocuparse de su creación y/o eliminación. Este proceso da como resultado inserciones escritas en sql para una base de datos mysql en un archivo por cada hora analizada.

- *extractor de visitas por día* La información fue recopilada por medio del servició Wikipedia article traffic statistics (`http://stats.grok.se`) que recopila las visualizaciones de los artículos agrupados por día. El servicio genera la información en dos representaciones, html mostrando un gráfica de barras de la distribución de las visualizaciones en un mes y en json mostrando la misma información por un mes. Al momento de realizar una petición del análisis de una consulta el analizador llama al

extractor por día para extraer toda la información utilizando esta página como fuente realizando un petición por cada mes del año. Este proceso es lento pero no tanto como la extracción por hora por lo que el sistema puede analizar cualquier página que se desee gracias a esta información.

- **Visualizador** Es una aplicación cliente que permite analizar consultas de la wikipedia. Se utilizó angularjs porque posee muchas utilerias para la interacción con un servidor REST y ofrece una base para la división de la aplicación y su componentes visuales en módulos individuales con funciones muy especificas. Se escribieron 2 módulos principales.
  - *d3-time-line* Este módulo tiene el objetivo de representar la duración de las revisiones de un artículo marcando el inicio del vandalismo en un serie de tiempo. El módulo se alimenta del primer servicio de los analizadores. (Se detallara mas su funcionamiento en el siguiente apartado)
  - *d3-affected* Este módulo es alimentado por el segundo servició visualizando los accesos que hay al artículo en el periodo de tiempo que se desea analizar.

Se utilizaron dos técnicas de visualización un por cada módulo descrito anteriormente.

- **Visualizar la duración**. Para representar la duración se gráfico la longitud del número de bytes que posee cada edición por medio de una gráfica de área donde la altura esta dada por el número de bytes del contenido de cada edición. El eje de las ordenadas es la longitud en bytes del artículo. Se utilizo colores para visualizar la duración de las versiones , cuando una versión es reemplazada por la siguiente. Se agrego una capa superior al gráfico representado el vandalismo por medio de rectángulo que su ancho es de la longitud del vandalismo en el tiempo, sin embargo, hay vandalismo que su ancho es menor a un pixel por lo que para estos casos se aproximo a 1 pixel el ancho de la recta. Se utilizo la propiedad alfa del área de la recta para identificar si hay vandalismos traslapados entre ellos. (Ver figura9). Los colores de las ediciones son aplicados por medio de aplicar la técnica de coloración colorbrewer inventada por Cynthia brewer para la visualización de atributos dentro de un mapa (ejemplo: la densidad poblacional, el nivel de educación, etc) (`http://www.personal.psu.edu/faculty/c/a/cab38/`). Se utilizo esta técnica para diferenciar cuando empieza y termina una edición.
- **Visualizar el daño**. Muestra por día los accesos realizados en la ventana de tiempo que se visualiza. Se selecciono este tipo de gráfica para no visualizar tendencias en las visitas si no grupos separados según la granuralidad de la información (Ver figura 10) para visualizar como afecta la granuralidad de la información al momento de saber cuanto ha afectado el vandalismo. Se complemento agregando lineas de control para indicar el cambio de una edición y se agrego barras rojas del mismo tamaño indicando que hay un porcentaje del fragmento del tiempo en vandalismo. Se utilizo gradientes de colores para indicar la magnitud de los accesos como también la altura de la barra, permitiendo ver a simple vista la distribución de las visualizaciones en todo el rango de tiempo que se desea analizar.

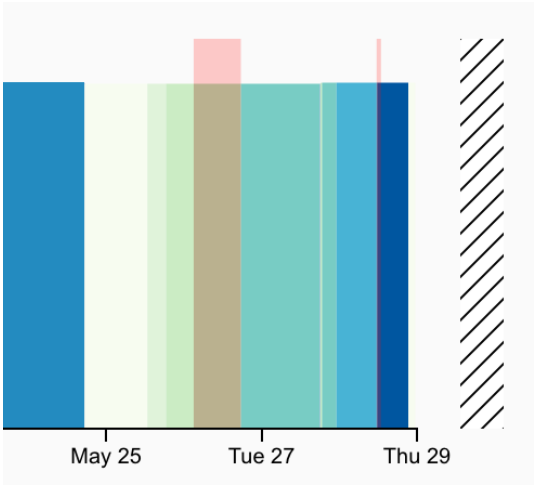


Figura 9: Representación de dos eventos de reversión por vandalismo representados en color rojo. El gradiente de colores entre azul y verde representa la duración de cada edición en el tiempo

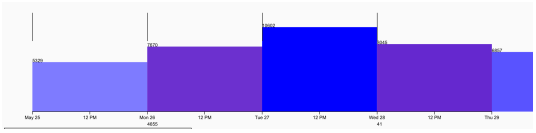


Figura 10: Representación de los accesos por día entre mayo 25 y 29 para el artículo *Madrid*

La interfaz se compone de una ventana para ingresar el rango de fechas a analizar y los títulos de la páginas que se desean visualizar porque la aplicación esta diseñada para analizar mas de un artículo al mismo tiempo para comparar resultados y tendencias.

Los datos extraídos fueron guardados en una base de datos mysql que posteriormente se conecto a *Rstudio* para el análisis posterior de la data utilizando ggplot para gráficar las tendencias de los valores.

6. Análisis Vertical y Horizontal

Se realizaron dos tipos de análisis uno vertical y otro horizontal. Vertical se refiere a un análisis individual de ciertas artículos para explorar el daño provocado por el vandalismo de forma particular. Y horizontal es analizar el daño de la Wikipedia provocado por el vandalismo para todos los artículos en una ventana de tiempo (un mes, una semana, un día).

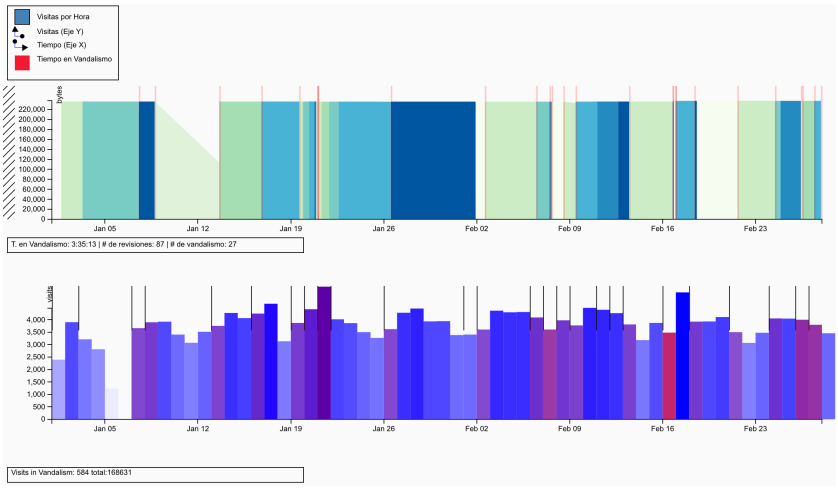


Figura 11: Interfaz gráfica que muestra las dos técnicas de visualización, el resumen de los datos del tiempo en estado de vandalismo y cuantas visitas en total fueron afectadas para el artículo *Madrid*

6.1. Resultados:Vertical

Se analizaron 300 páginas para comprobar el funcionamiento de la herramienta entre ellas se seleccionaron 3 que ofrecen características interesantes por el comportamiento de las ediciones

- **Venezuela** título del artículo que habla sobre el país de Venezuela. tiene 164 usuarios colaboraron en su edición en todo el año (30 bots, 5 administradores, 53 usuarios registrados y 76 anónimos). En el año Venezuela sufrió 76 reversiones por vandalismo con una duración de 11 días, 7 horas con 52 minutos, en total se afectaron 45,416 visualizaciones de un total en el año de 2,079,230 que representa el 2 % de todas las visualizaciones del año. (Ver figura **??**).

Club América título del artículo que habla sobre el club profesional de futball ubicado en México. El total de tiempo en vandalismo son 5 días con 21 horas y 22 minutos. Calculando un total de 5660 visitas en vandalismo que representa el 1 % de las visitas al artículo.

Dragon Ball título del artículos de la serie de televisión japonesa. Posee 98 revisiones de vandalismo de un total de 475 durando 18 días 13 horas y 18 minutos, si lo comparamos con el articulo del Çlub América"que posee 1,514 revisiones y solo 54 son vandalismo siendo 5 días 21 horas en vandalismo identificamos que el número de ediciones no es un dato que asegura la duración del vandalismo siendo el presente artículo mas afectado. Este artículo presenta un fenómeno que al momento de guardarse los datos no se persistieron y por lo

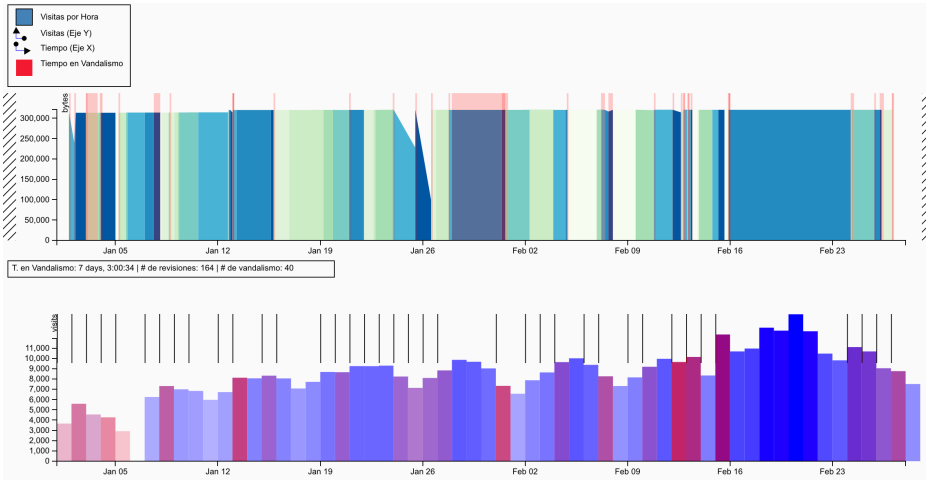


Figura 12: Serie de tiempo del mes de enero 2014 del cambio de bytes en el contenido del documento *Venezuela*

tanto no se pudieron recuperar por lo que para esa revisión en concreto no se sabe su longitud (Ver figura **??**).

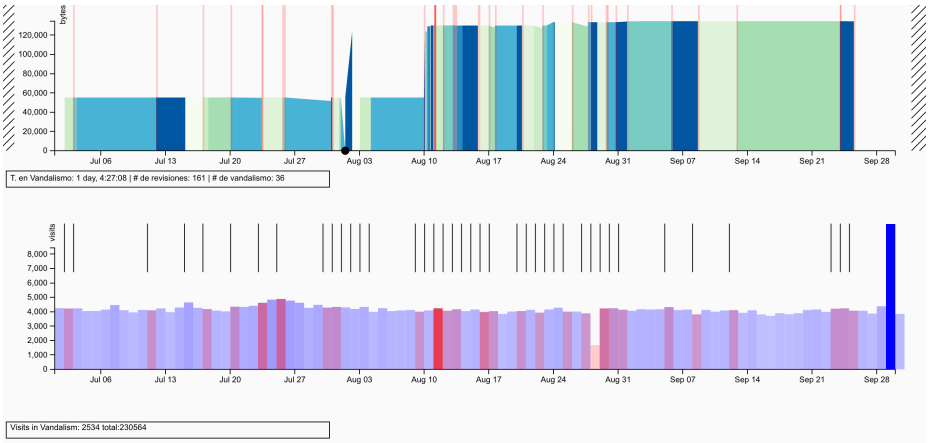


Figura 13: Serie de tiempo del mes de agosto y septiembre del año 2014 para el artículo *Dragon Ball*, el punto negro indica el desconocimiento de la longitud de la revisión

6.2. Discusión de Resultados:Verticales

Como podemos observar en los tres ejemplos anteriores la meta-data de una edición es suficiente información para el análisis del impacto del vandalismo como sugiere Priedhorsky. también podemos ver ráfagas de edición observando que la lineas de las revisiones están mas cercanas unas de otras estudiadas por MirkoKampf que afirma que no hay ninguna relación con el número de visitas y la ediciones acercándose más a ruido blanco. Estos datos fueron calculados utilizando la suma de las visitas por día de cada artículo por lo tanto no se tiene la distribución real de las visitas. Se analizaron las 300 páginas más revertidas y se encontró que el mayor tiempo que ha estado una página en estado de vandalismo no pertenece a las mas editadas ni las mas revertidas si no a *Potenciación* teniendo un vandalismo de 212 días afectando 383,171 visualizaciones que son el 60 % de las visitas en un año donde se comprobó que pertenece a una movimiento entre páginas; se comprobó que la reversión de potenciación por la manera en que se comporta la reversión porque si es una reversión de mucho tiempo (202 días) es porque se regreso el contenido a un estado dividiendo el contenido en más páginas. También las pruebas realizadas el número de visualizaciones afectadas no llega a ser mas del 5 % si se eliminan estos casos especiales lo que nos indica que en un año en promedio se afectan 7 días. El aplicativo desarrollado ha demostrado ser práctico para la medición de las visualizaciones afectadas, sin embargo, sin un análisis profundo del contenido que existía en cada una de la revisiones no hay forma alguna de saber el origen de la decisión de la reversión, punto que seria de interés para futuras estadísticas.

6.3. Resultados:Horizontales

Para el análisis horizontal se escogió el mes de diciembre por ser el mes más cercano a la actualizad, este análisis se realizó con la información por hora de las visualizaciones de los artículos por ofrecer mayor detalle. En el mes las páginas con más impacto por el vandalismo es *Nochebuena*, *Cuento* y *Familia* (32,284, 26,938, 16,894 visualizaciones respectivamente). En total hay 2,316,807 consultas en estado de vandalismo de un total para el mes de 350,481,966 siendo menos del 1 % de consultas en el año.

Agrupando las horas del día se puede observar que el vandalismo se da mas en ciertas horas que otras. Para el mes de diciembre a partir de las 12 de tarde hasta las 10 de la noche hay mas actos de vandalismo (Ver figura 14) . Si lo comparamos con la gráfica de los accesos por hora del día vemos que es la horas del día con mayor número de visualizaciones de artículos (Ver figura. 16). En la gráfica 15 podemos observar una tendencia entre mas impacto genera una revisión mas duración tiene el vandalismo. Sin embargo hay artículos que no siguen esa tendencia como son *Cuento* y *Nochebuena* que su vandalismo dura menos tiempo pero su impacto es muy grande. O como *Premios_ Nuestra_ Tierra* Que ha tenido un vandalismo de muy poco impacto pero ha durado mucho tiempo.

La Figura 18 nos muestra un punto por cada usuario rever-sor que repara a Wikipedia. Podemos observar que normalmente un usuario su contribución al

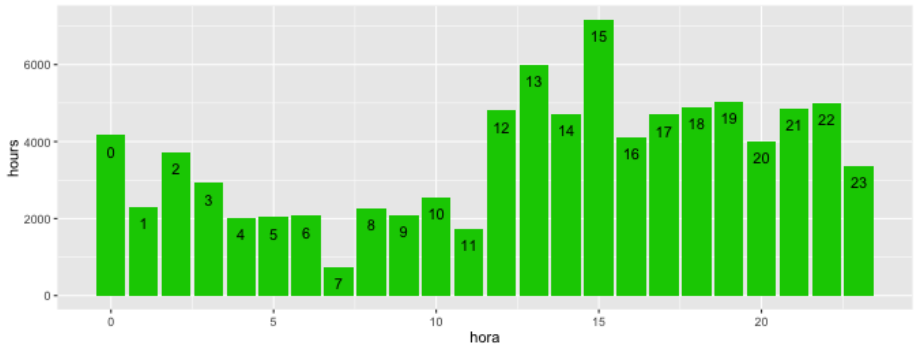


Figura 14: Gráfica de la horas del día y el impacto del vandalismo, los números son las horas del día en formato 24horas

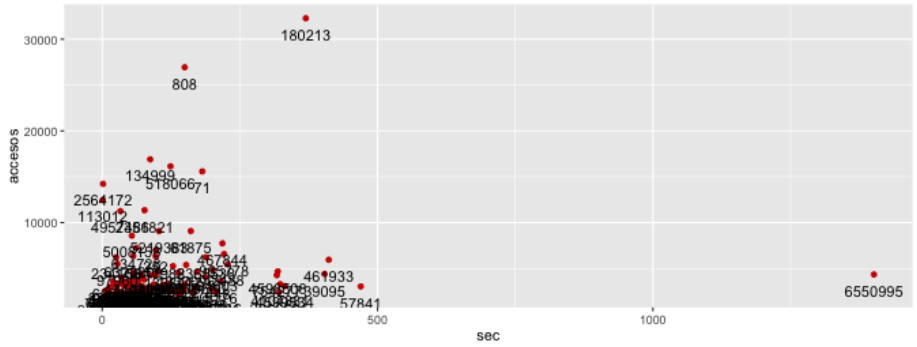


Figura 15: Gráfica accesos vrs duración del vandalismo, el punto 6550995 pertenece a Premios_Nuestra_Tierra y el punto 808 y 180213 pertenecen a Cuento y Noche buena, todos los demás artículos se mantienen con muy poco impacto y una duración promedio. Nota. La duración del vandalismo puede ser mayor a la de un mes porque se contabiliza la duración total desde que empieza el vandalismo que puede ser antes de diciembre para medir correctamente el impacto histórico de la reversión

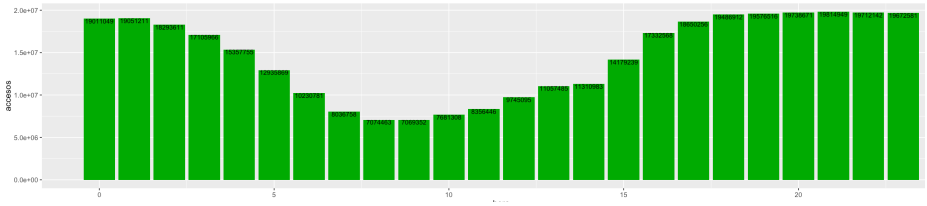


Figura 16: Accesos totales por hora del mes de diciembre, cada barra representa una hora del día

ataque contra el vandalismo es de muy poco impacto por otro lado hay usuarios excepcionales que contribuido con la reparación de Wikipedia a lo largo del mes y han sido los que mas impacto han revertido (Eduardosalg, Gusama Romero los dos puntos en azul al final después de la marca 150,000 de accesos). Los robots en Wikipedia forman la primera linea de defensa contra el vandalismo siendo el mas eficiente *patruRobot* que ha sido el robot que mas ha reparado y que menos impacto a creado. Si analizamos ahora las personas que provocan vandalismo (Ver figura 19) vemos que la mayoría de usuarios realizan muy pocas ediciones y los usuarios que si tienen muchas ediciones de vandalismo produce muy poco impacto. En cambió el usuario que ha dañado más a Wikipedia es una Ip anónima de México con un impacto de 94,525 visitas (201.102.47.14).

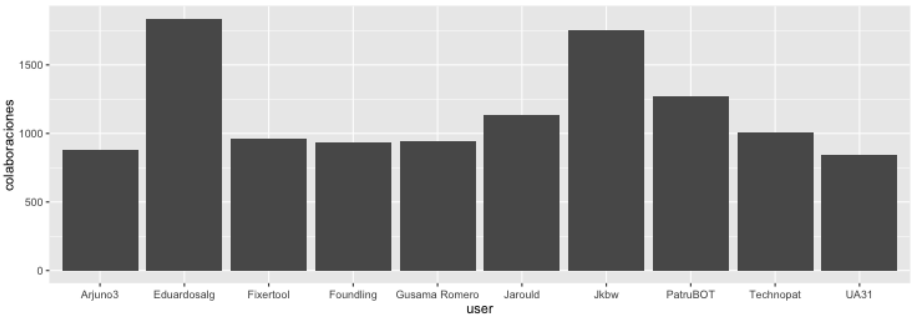
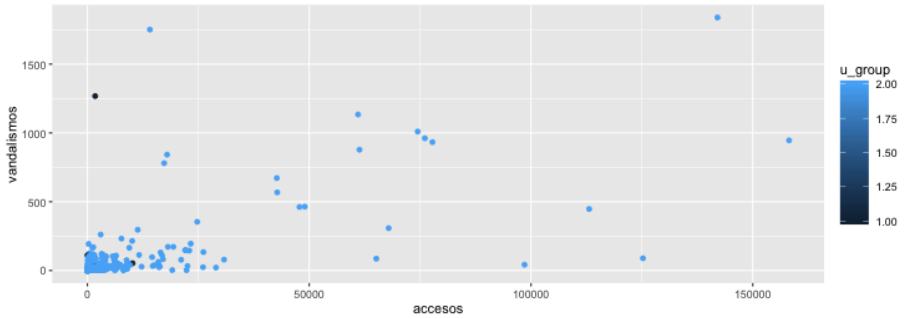


Figura 17: Usuarios que mas vandalismo reparan

6.4. Discusión de Resultados:Horizontales

La gráfica 19 es una ayuda visual importante mostrando la distribución del comportamiento de la gente que provoca reversiones por vandalismo y su impacto en las visualizaciones, porque entre mas cerca este de cero en el eje de las abscisas menos ha afectado, también podemos observar que la mayoría de vándalos provocan un solo acto de vandalismo y que los que provocan muchos actos continuamente son detectados con relativa rapidez y no crean mucho impacto porque entre mas alejado este el punto de las ordenas mas actos de vandalismo ha ejecutado. Este comportamiento demuestra que los usuarios que realizan vandalismo continuamente son controlados por los mismos controles internos de la Wikipedia evitando que sigan creando problemas. Otro dato interesante es la aparición de los usuarios administradores a este gráfica , porque sus actos presunto vandalismo son de hecho actos que pertenecen a otros efectos (fusión de páginas, movimiento de páginas).

La gráfica 18 muestra de forma intuitiva el desempeño de los usuarios reversores. Si tomamos por ejemplo los programas automáticos o bots vemos que entre mas cerca estén del cero en el eje de las abscisas y mas alejado de las



- **azúl claro** representa todos los usuarios administradores
- **azúl oscuro** representa todos los programas automáticos

Figura 18: Gráfica número de vandalismos reparados versus la suma del impacto o consultas reparadas. Nota. La duración del vandalismo puede ser mayor a la de un mes porque se contabiliza la duración total desde que empieza el vandalismo que puede ser antes de diciembre para medir correctamente el impacto histórico de la reversión

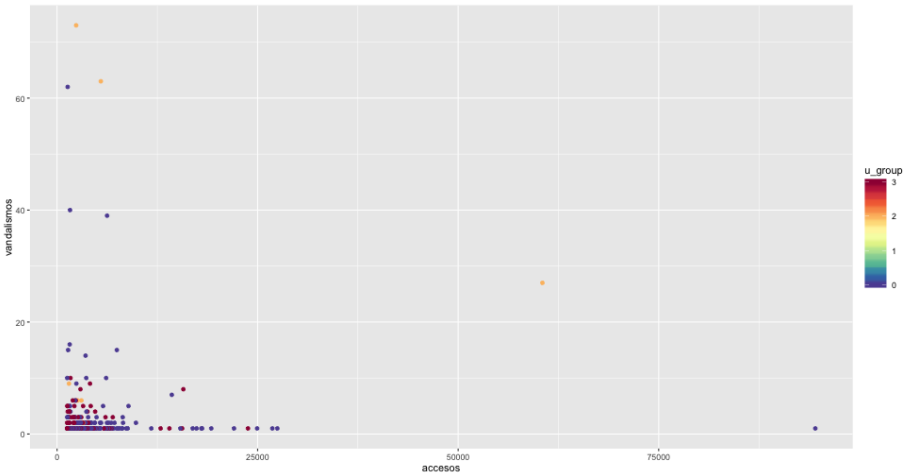


Figura 19: Gráfica número de vandalismos versus la suma del impacto o consultas de los usuarios que provocan vandalismo

- **amarillo** usuarios administradores
- **rojo** usuarios registrados
- **azul** usuarios anónimos

ordenadas es mejor en su tarea porque quiere decir que repara rápidamente y por lo tanto no se afectan muchas visualización a las páginas. En cambio vemos

los usuario *clave* para la estabilidad de los artículos donde entre mas alejados estén del cero en el eje de las ordenadas mas trabajo han realizado pero lo interesante es el impacto que han reparado al hacer el trabajo por lo que se puede observar usuarios que han revertido contenido con mucho tiempo de exposición. Este fenómeno mas se identifica con otros efectos y no de vandalismo (fusión de páginas, guerras de ediciones etc).

7. Conclusiones

El vandalismo es fenómeno de mucho interés para el investigador de la Wikipedia por las posibles repercusiones en el mundo real. En el presente trabajo se manipulo información de varias fuentes heterogéneas (bases de datos, ficheros de datos, servicios rest) por medio de varios scripts de transformación a una estructura común para su análisis posterior porque el volumen de datos a analizar ha demostrado ser un reto de carga computacional considerable. También se creo un servicio de transformación de datos que analiza en tiempo real el vandalismo dentro de un rango de tiempo. Este servicio fue concebido con relativa facilidad por la ventaja del lenguaje de programación que se utilizo al posee tipos de datos específicos para fragmentos de tiempo. Se ha presentado un sistema de análisis visual para el efecto del vandalismo en los artículos de Wikipedia que han sido vandalisados basado en técnicas de visualización clásicas y modernas. Se identificaron otros efectos de las reversiones como es la guerra de ediciones o la fusión de páginas estas pueden confundirse con el vandalismo pero a pesar de ello la técnica visual da una base solida para el entendimiento del efecto del vandalismo. Se ha calculado de forma eficiente el impacto del vandalismo por hora y por día utilizando un algoritmo que calcula la proporción del vandalismo en el fragmento de tiempo, sin embargo esta medición se basa en que todas visualizaciones son constantes en ese periodo de tiempo afirmación que no es cierta pero para fines prácticos ha demostrado ser una medida muy valiosa en la medición del impacto del vandalismo. Se demostró que la Wikipedia posee un sistema muy efectivo contra el vandalismo donde una persona que provoca mucho vandalismo constantemente es controlada con relativa rapidez evitando que el daño continue.

Como punto final decir que el uso de técnicas de visualización provee al usuario de más inteligencia en la toma de decisiones y en comprender problemas mas complejos de forma intuitiva y rápida; lo anterior en mi opinión debe ser una de las primeras opciones para la comprensión de problemas que abarcan muchos datos por la disminución de la carga cognitiva a los usuarios.

Referencias

1. Shun-Ling Chen. Self-governing online communities in web 2.0: Privacy, anonymity and accountability in wikipedia. *Alb. LJ Sci. & Tech.*, 20:421, 2010.
2. Maxime Clement and Matthieu J Guitton. Interacting with bots online: Users reactions to actions of automated programs in wikipedia, 2015.

3. Manoj Harpalani, Thanadit Phumprao, Megha Bassi, Michael Hart, and Rob Johnson. Wiki vandalysis-wikipedia vandalism analysis-lab report for pan at clef 2010. In *CLEF (Notebook Papers/LABs/Workshops)*, 2010.
4. Michał Jankowski-Lorek, Szymon Jaroszewicz, Łukasz Ostrowski, and Adam Wierzbicki. Verifying social network models of wikipedia knowledge community. *Information Sciences*, 339:158–174, 2016.
5. Mirko Kämpf, Sebastian Tismer, Jan W Kantelhardt, and Lev Muchnik. Fluctuations in wikipedia access-rate and edit-event data. *Physica A: Statistical Mechanics and its Applications*, 391(23):6101–6111, 2012.
6. Reid Priedhorsky, Jilin Chen, Shyong Tony K Lam, Katherine Panciera, Loren Terveen, and John Riedl. Creating, destroying, and restoring value in wikipedia. In *Proceedings of the 2007 international ACM conference on Supporting group work*, pages 259–268. ACM, 2007.
7. Ángel F Zazo Rodriguez, Carlos G Figuerola, and José Luis Alonso Berrocal. Edición de contenidos en un entorno colaborativo: el caso de la wikipedia en español. *Scire: representación y organización del conocimiento*, 21(2):57–67, 2015.
8. Koen Smets, Bart Goethals, and Brigitte Verdonk. Automatic vandalism detection in wikipedia: Towards a machine learning approach. In *AAAI workshop on Wikipedia and artificial intelligence: An Evolving Synergy*, pages 43–48, 2008.
9. Khoi-Nguyen Tran, Peter Christen, Scott Sanner, and Lexing Xie. Context-aware detection of sneaky vandalism on wikipedia across multiple languages. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 380–391. Springer, 2015.
10. Fernanda B Viégas, Martin Wattenberg, and Kushal Dave. Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 575–582. ACM, 2004.
11. Taha Yasseri and János Kertész. Value production in a collaborative environment. *Journal of Statistical Physics*, 151(3-4):414–439, 2013.

Desarrollo de robots imprimibles en entornos educativos

Alberto Encinas Elvira, Vidal Moreno Rodilla, Belén Curto Diego y Francisco Javier Blanco Rodríguez

Máster en Sistemas Inteligentes
Universidad de Salamanca

Resumen Este documento trata de la realización de un robot destinado a la educación y que pueda ser utilizado en cualquier nivel educativo para fomentar los conocimientos STEM (Science, Technology, Engineering and Mathematics). Es un robot modular, basado en protocolos abiertos de forma que cualquiera pueda ampliar las características del mismo diseñando e imprimiendo módulos personalizados. Su gran versatilidad permitirá su utilización con varias placas microcontroladoras y software de programación por lo que su uso puede ampliarse a muchos niveles educativos. Debido a que la robótica es una rama multidisciplinar que combina conocimientos de matemáticas, física, electricidad y electrónica, informática, control automático, mecánica, etc la plataforma puede ser orientada al estudio más profundo de cada una de estas ramas lo que extiende aún más su campo de aplicación e interés.

Keywords: Raspberry, impresión 3D, robot, Open Structures, control, navegación.

1. Introducción

Desde siempre el ser humano ha querido tener a algo o alguien que le facilite las tareas más arduas, difíciles y monótonas de su vida cotidiana para poder dedicarse a aquellas actividades que le realizan como persona. Son muchas las actividades que nadie quiere realizar y desde hace tiempo las personas con mayor poder adquisitivo y clase social más alta han buscado, desde esclavos en la edad antigua y media, hasta asistentes en la actualidad que se encarguen de esas tareas. Con el fin de eliminar de nuestras tareas diarias estas actividades repetitivas y aburridas la sociedad ha demandado robots que realicen por nosotros esas tareas. Al área de la robótica que se ocupa de investigar y desarrollar este tipo de robots que nos ayudan en tareas cotidianas se ha denominado "robótica de servicios" [1] y abarca desde los robots de limpieza como Roomba [1] , robots guía en museos u otras exhibiciones, en puntos de información, robots de videovigilancia doméstica, robots de asistencia en hoteles [10]...

[1] Página web de Roomba: http://www.irobot.es/robots-domesticos/aspiracion

Dentro de la "robótica de servicios", un área que está experimentando un interés creciente en acercar la robótica a todos los públicos exponiendo las grandes ventajas de este campo y han surgido lo que se conocen como plataformas educativas de robótica. Estas plataformas están pensadas para enseñar a cualquier tipo de público, desde los más pequeños hasta gente más experimentada, las posibilidades que abren los robots en el mundo.

Entre las ventajas, se encuentra utilizar la robótica en la educación, con el fin de potenciar las áreas de conocimiento conocidas por STEM (Science, Technology, Engineering and Mathematics) [3][9]. Es por esto por lo que muchas han sido las empresas que han decidido investigar y adentrarse en el campo de la robótica educativa y ofertan plataformas de trabajo que facilitan la tarea de crear un robot en el aula, en nuestra casa... Además, la robótica educativa potencia también las actividades cognitivas y sociales de los jóvenes como son por ejemplo la resolución de problemas, el pensamiento creativo e innovador y las habilidades de investigación [5][6][7]. También fomenta un aprendizaje actitudinal y sirve para impulsar un mayor interés por la ciencia y la tecnología debido a la escasez de estudiantes en estas ramas y la creciente demanda de gente preparada en este campo [8].

Con la robótica educativa se consigue dar un enfoque más práctico a materias como la física o las matemáticas. De esta forma se consigue pasar del enfoque tradicional de resolver las ecuaciones del libro sin aplicar a ningún caso en particular a ver realmente la utilidad de esas ecuaciones y demás procedimientos matemáticos en la vida real.

La realización de un curso de robótica de 4 meses de duración ha servido para demostrar que se produce ese aumento en la motivación de los alumnos. En el curso de robótica participaron un total de 27 alumnos con edades comprendidas entre 14 y 16 años. A lo largo del curso se propusieron actividades de programación, diseño de figuras 3D y electrónica que permitían fomentar estos conocimientos explicados con anterioridad. El resultado de las encuestas que rellenaron los chicos al final del curso relevó que efectivamente se produce un aumento de interés, por gran parte de los alumnos, en campos como la programación o la electrónica y se refuerzan las bases que han estudiado en el instituto en asignaturas como matemáticas y tecnología, ya que emplearon principios de estas asignaturas para resolver las diferentes actividades que se propusieron. Además, los alumnos se dieron cuenta de que la robótica es la integración de muchas otras áreas de conocimiento y vieron cómo se aplicaban conceptos estudiados en clase de física o música al funcionamiento de determinados sensores y actuadores, lo que supuso una gran sorpresa para algunos de ellos, ya que no esperaban que los conceptos que les enseñaban en clase de música en el instituto estuviesen relacionados con algo que creían totalmente diferente como la robótica.

Es por todas estas características, por lo que la robótica educativa despertando un alto interés en los centros de educación y por lo que he decidido enfocarme

en esta área y realizar un estudio de las posibilidades que hay disponibles actualmente para introducir la robótica educativa en los centros.

En este trabajo se va a desarrollar una plataforma robótica modular orientada al ámbito de la educación que pueda ser utilizada en varios niveles educativos para enseñar los diferentes campos que incluye la robótica como son las matemáticas, la mecánica, el control, etc [9]. Además de ser modular, las piezas con las que se construye el robot serán imprimibles prácticamente en su totalidad (excluyendo componentes eléctricos, electrónicos y aquellos elementos mecánicos que necesiten unas características mecánicas que no puedan ser obtenidas con el material de impresión disponible, como puedan ser ejes sometidos a grandes fuerzas de flexión, alambres, cables flexibles o cadenas sometidos a fuerzas de tracción, etc). Dado el carácter imprimible de la plataforma se podrá conseguir que los alumnos desarrollen capacidades adicionales comoson el análisis, creatividad y diseño CAD en 3 dimensiones [9] para poder generar ellos mismos piezas únicas para el robot, eliminando así las posibles restricciones de elementos mecánicos.

En la elaboración del trabajo se ha realizado un análisis de los kits de robótica educativa que existen actualmente en el mercado, centrándome en los dos de mayor impacto en nuestro país: LEGO [2] y bq [3]. Para cada uno de los kits analizados se han visto las carencias que tienen para tratar de resolverlas con la plataforma propuesta.

1.1. Kits comerciales de robótica educativa

En la actualidad existen varios kits de robótica que son utilizados en el campo de la educación, pero en España destacan sobre todo los kits que ofrecen las empresas de LEGO y bq. Cada kit está formado por dos componentes principales y básicos en este tipo de kit que son los elemento hardware, con los que se crea el robot y un entorno de programación donde se desarrollan los programas del robot. Si bien ambos kits cuentan con estos componentes, cada empresa tiene un enfoque diferente y cada kit tiene, por tanto, carencias diferentes.
Realizando un estudio de ambos kits, se llega a la conclusión de que la principal carencia de los kits de LEGO es la poca variedad de sensores, el alto precio que hay que pagar por el kit y por cada sensor extra y la limitación por parte del módulo central de ofrecer únicamente 4 puertos de entrada y otros 4 puertos de salida, limitando así la cantidad de elementos (sensores y actuadores) que se pueden conectar al módulo.
En el caso del kit de bq se suple esta carencia, ya que posee una gran variedad de sensores y actuadores diferentes y la placa controladora posee una mayor cantidad de entradas y salidas, pero en este caso no se dispone de material estructural

[2] Página web de LEGO Mindstorms EV3: http://www.lego.com/es-es/mindstorms
[3] Página web del kit ZUM de bq: https://www.bq.com/es/zum-kit

que permita dar forma a las creaciones. Todo se limita a conectar los sensores a la placa y ver cómo funciona. Una vez analizadas las carencias de ambos kits se buscará un sistema que trate de eliminarlas.

2. Desarrollo de la propuesta

Lo que se propone es la realización de un robot que pueda ser utilizado en un ámbito educativo con las mínimas restricciones posibles en cuanto al nivel educacional de los usuarios, permitiendo así su uso desde los niveles más básicos de conocimiento hasta los más avanzados.
La plataforma es modular e imprimible, de forma que cualquier usuario sea capaz de desarrollar sus propias piezas, permitiendo así un mayor desarrollo de la creatividad e imaginación de los usuarios. Esto ofrece la posibilidad de trabajar otras áreas como el diseño y la impresión 3D.
Para el correcto desarrollo del robot se siguieron diferentes etapas: En primer lugar, se hizo un estudio bibliográfico previo y realizó un estudio de los elementos (tanto sensores como actuadores) que se utilizarían en la elaboración del robot. En segundo lugar, se realizó un esbozo del robot a desarrollar y se hizo un planteamiento de la ubicación de cada uno de los elementos previamente seleccionados. En tercer lugar, se realizó el diseño de todas las piezas que formaban el robot y se realizaron las simulaciones pertinentes para garantizar la ausencia de colisiones o incompatibilidades antes de imprimirlas. En cuarto lugar, con la estructura impresa se realizan simulaciones de los sistemas electrónicos y una vez verificado su correcto funcionamiento en las simulaciones se adquieren los componentes y se montan en la estructura anterior. Para finalizar, se cargan los programas desarrollados de prueba, se realizan los últimos ajustes y se comprueba el correcto funcionamiento.
Durante la etapa de estudio bibliográfico previo y estudio de los sensores se estudiaron los kits actuales y su modo de funcionamiento, los sensores y actuadores de los que disponían y se vieron las carencias que disponían para intentar suplirlas con la nueva plataforma. Aquí también se buscó información acerca de la modularidad de los sistemas, viendo las diferentes formas de hacer sistemas modulares, estudiando diferentes posibilidades y seleccionando aquella que proporcionaría una mayor sencillez y facilidad de uso y montaje del robot. La elegida fue la denominada "Open Structures", cuyo funcionamiento es análogo al que se basa "Meccano" y cuya plantilla de diseño aparece en la Fig. 1. Consiste en piezas perforadas diseñadas siguiendo un patrón de orificios de forma que al superponer unas sobre otras se hace coincidir los orificios de ambas y se unen por medio de elementos de unión pasantes como los conjuntos tornillo-tuerca empleados en la unión de los elementos de este proyecto.

Durante la segunda etapa se planificó la estructura del robot que se iba a realizar teniendo en cuenta el método modular elegido en la fase anterior. Este modelo fija determinados parámetros de las piezas ya que todas deben construirse siguiendo el mismo patrón para garantizar la coincidencia de los orificios

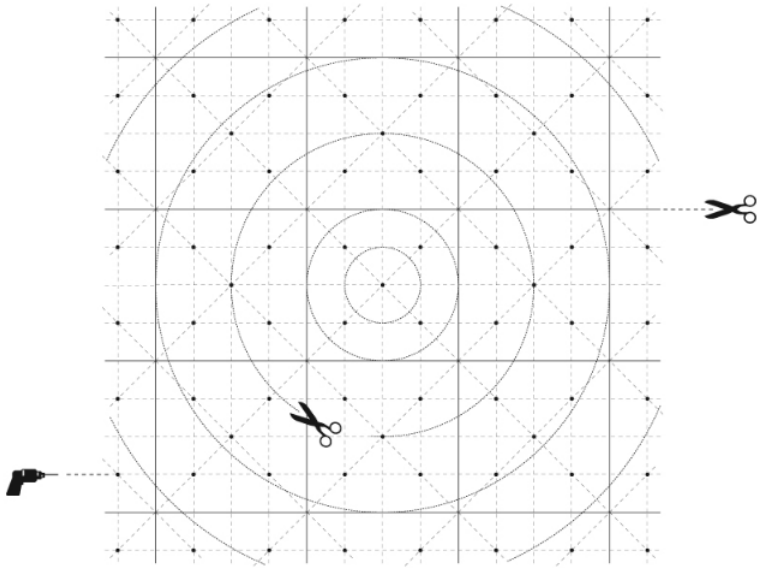


Figura 1: OS Grid de OpenStructures

una vez superpuestas. Cada pieza se diseñó de forma que pudiese tener al menos cuatro puntos de unión con otra. Como base del robot se empleó una plancha cuadrada de 16cm x 20cm que sirviese como nexo entre las diferentes piezas y confiriese firmeza al conjunto. Un ejemplo de esta práctica se muestra en la Fig. 2 En esta etapa también se definieron aspectos como la configuración motriz del robot y la cámara, optando para el robot por una configuración diferencial empleando dos ruedas motrices y una rueda loca y para la cámara una opción de "pan and tilt" permitiendo el movimiento en horizontal y vertical. Esto supuso el diseño de una nueva estructura para la cámara que permitiese realizar estos movimientos y se agregó a dicha estructura soporte para ocho LEDs con el fin de dotar una fuente de luz para poder utilizarla en condiciones de baja iluminación.

En la tercera etapa se diseñaron por ordenador todas las piezas esbozadas en la etapa anterior, modificándolas lo necesario para evitar conflictos tanto a la hora del montaje como posibles situaciones complicadas durante la impresión de las mismas. El diseño se hizo buscando siempre una reducción del material extruido por la impresora y consiguiendo por tanto tiempos de impresión menores en todos los casos posibles. Antes de realizar la impresión de cada pieza se comprobó su correcta coincidencia con el resto de elementos a los que iba unida y comprobando que su colocación era físicamente posible evitando así que hubiese que repetir piezas debido a una mala planificación, colisiones con otras

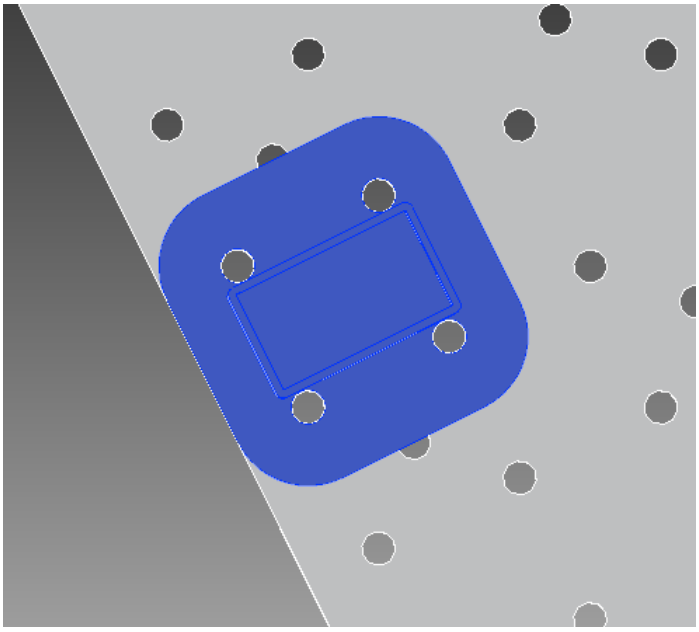


Figura 2: Ensamblaje de piezas utilizando OpenStrucutres

piezas o imposibilidad de colocarlas por falta de espacio. Una vez comprobada cada pieza se pasaba a la impresión de la misma.
En la cuarta etapa se realizó el diseño electrónico de los circuitos en el ordenador y se comprobó el correcto funcionamiento del sistema. En este proceso se incluye el cálculo de resistencias y demás elementos electrónicos para garantizar un funcionamiento correcto y dentro de los límites de operación de cada elemento. También se calcularon los requerimientos de batería del total de los conjuntos con el fin de elegir la más adecuada. Una vez comprobado que todos los circuitos funcionan de manera correcta con los parámetros calculados en la simulación por computador se hizo el presupuesto de los elementos utilizados y tras su aprobación se realizó el pedido de todos los componentes necesarios.
Por último, se desarrollaron algunos programas de prueba con el fin de poder ver el comportamiento del robot con cada sensor. Para ello se hicieron programas para probar el movimiento del robot y los sistemas de seguridad implementados con sensores de distancia, la cámara y la correcta transmisión del vídeo a través de internet y el funcionamiento del movimiento en horizontal y vertical de la cámara, así como el encendido y apagado de la luz de cámara. Primero se probaron los programas por separado empleando únicamente el sensor que empleaban navegación para verificar el correcto funcionamiento. Una vez que todos los programas por separado producían el resultado esperado se pasó a juntar aquellos programas que debían funcionar de manera conjunta, como es el caso del sensor de distancia y la navegación. Durante el movimiento del robot, el sensor de dis-

tancia debe tomar medidas para evitar colisionar con algún elemento.

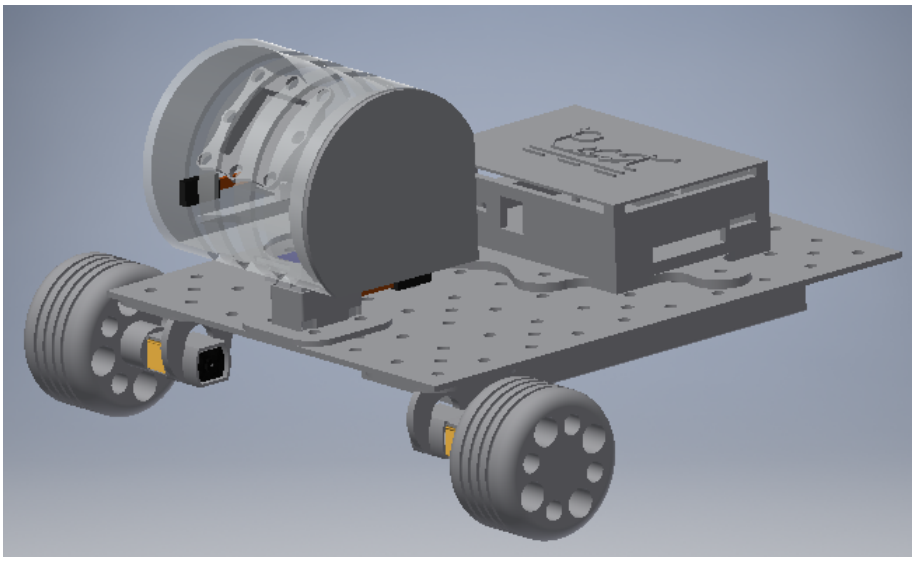


Figura 3: Robot completo

3. Análisis de los resultados

Tras haber implementado algunos programas de prueba para comprobar el funcionamiento del robot en su conjunto se ha visto que es el correcto, aunque se presentan algunas limitaciones que se deben a la placa controladora. Estas limitaciones están relacionadas generalmente con restricciones temporales y se limitan únicamente a aquellas tareas que tienen unos requisitos de tiempo muy estrictos como es el caso de los servo motores. Para realizar el control de este tipo de motores se han de generar pulsos de tensión entre 900 microsegundos y 2100 microsegundos, asignándole a cada duración una posición determinada del eje del motor, siendo un pulso de 900 microsegundos una de las posiciones extremas del eje, la que se correspondería con un ángulo de 0^{o} o referencia, y un pulso de 2100 microsegundos la correspondiente al extremo opuesto, que, para los motores elegidos es de 180^{o} aproximadamente. De esta forma, un retardo de apenas 6,6 microsegundos se correspondería con un desfase de 1^{o} en la posición del eje. Estos requisitos temporales tan ajustados, en los que no se puede tener un retardo de 6,6 microsegundos en la acción de control, no pueden ser cumplidas por la placa empleada ya que genera inestabilidad en la posición del eje apareciendo vibraciones que empeoran la calidad de las imágenes tomadas por

la cámara.
Para solucionar este problema se pueden proponer varias opciones. Una de ellas sería la sustitución de la placa por otra que cumpla estos requisitos temporales. Algunas de placas que han demostrado un buen funcionamiento en esta situación son las placas Arduino y sus derivadas, que al carecer de sistema operativo proporcionan un mayor control. Esta tarea es bastante fácil de realizar y apenas requiere esfuerzo ya que se pueden aprovechar prácticamente todos los módulos empleados y sólo habría que traducir el código de cada tarea. En caso de ser necesario el uso de un sistema operativo en la placa porque las tareas que realiza así lo requieren, sería posible utilizar un sistema operativo de tiempo real para este tipo de tareas y programar de forma adecuada la tarea de control de los servo motores.
Otra de las grandes ventajas que presenta esta plataforma es el amplio rango de la educación en el que se puede utilizar. Tras realizar los programas en Python y en Scratch se ha comprobado que se puede utilizar para enseñar los conceptos más básicos de robótica en los niveles educativos más básicos utilizando el conjunto de Arduino y S4A (Scratch for Arduino) y que se puede ampliar fácilmente cambiando la placa a una con mayores posibilidades como es Arduino MEGA. A medida que se va ascendiendo en niveles educativos se puede cambiar, además de la placa, el software de programación, pasando desde Scratch a Arduino IDE y en niveles ya superiores se puede emplear Python o C como lenguajes de programación para enseñar conceptos más avanzados. Además, se pueden utilizar las limitaciones temporales indicadas anteriormente o cualquier otra limitación que puede surgir al hacer algún proyecto con una placa y software determinados como herramienta para permitir que los alumnos se den cuenta de estas limitaciones y de la importancia de utilizar otras plataformas o herramientas o combinar varios sistemas con el fin de eliminar aquellas restricciones de una placa sin recortar las características del proyecto en su conjunto.

4. Conclusiones y Líneas futuras de mejora

En este artículo se ha indicado una plataforma de robótica educativa que trata de suplir las carencias de aquellas presentes en el mercado actualmente. Tras analizar los resultados se ha visto que la plataforma propuesta posee grandes mejoras respecto a las existentes actualmente como la posibilidad de utilizar la misma plataforma en niveles educativos muy diferentes. Es una plataforma altamente evolutiva, desde el punto de vista que cualquiera puede añadir los módulos nuevos y personalizados que se ajusten a los sensores y necesidades de cada uno, ya que es un diseño modular e imprimible. Todos estos factores favorecen por tanto el uso de esta plataforma en varios niveles educativos y en ese caso, reutilizar los sensores y módulos utilizados en un nivel en otro.
Otra ventaja de esta plataforma es la versatilidad ya que puede ser utilizada en todos los campos relacionados con la robótica como es la electrónica, la mecánica, la informática y el control, de forma que con una misma plataforma se pueda

utilizar para desarrollar prácticas en materia de control, mecánica, electrónica o incluso informática, dejando libertad para elegir el campo en el que se quiere centrar el desarrollo y uso de la plataforma.
A pesar de los buenos resultados obtenidos tras hacer varias pruebas se han detectado líneas de mejora que permitirían añadir aún más valor y funcionalidad a la plataforma. Una de estas mejoras posibles sería la creación de un repositorio de contenido en internet en el que existan módulos estándar compatibles con varios sensores y además la gente pueda subir las piezas y módulos que ha desarrollado para que puedan ser utilizadas, modificadas y mejoradas por otras personas con el fin de crear contenido de mayor calidad y facilitar el uso de esta plataforma a aquellas personas que no tienen muchos conocimientos en el diseño de piezas 3D o que quieren utilizar la plataforma centrándose en otros aspectos como puedan ser la informática o la electrónica.
Otro aspecto a destacar que se podría realizar como mejora en un futuro sería la inclusión de un software de visualización asociado al robot. Se ha pensado en desarrollar un programa de visualización que permita controlar al robot de manera remota con una interfaz gráfica amigable y que facilite además las tareas de supervisión y detección de posibles anomalías o fallos en el robot. Este software ayudará también a diseñar el robot ya que se podría trabajar con él en modo sin conexión y facilitaría la tarea de diseño hardware del robot al permitir al usuario añadir módulos al robot y visualizar, de una manera clara, el aumento de consumo que supondría la inclusión de dicho módulo en el robot, el número de entradas y salidas necesarias en la placa para controlar dicho módulo, así como el número de entradas y salidas libres disponibles en el módulo central. Una vez que el usuario esté conectado al robot, el software mostrará al usuario valores de interés como las medidas de los diferentes sensores que tiene conectados, el nivel de batería del robot o las imágenes que captura la cámara en caso de tener habilitada esta opción en el robot.

Referencias

1. Libro blanco de la robótica. (2008). [Madrid]: Comite Espanol de Automatica. [ultimo acceso: 05/06/2016] Disponible en: http://www.ceautomatica.es/sites/default/files/upload/10/files/LIBRO %20BLANCO %20DE %20LA %20ROBOTICA %202_v1.pdf

2. Jeschke, S., Volimer, U., Wilke, M. and Kato, A. (2008). Robotics in Academic Medical Engineering Education. 2008 IEEE/SICE International Symposium on System Integration. [último acceso: 05/06/2016] Disponible en: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4770426&newsearch=true&queryText=education %20robotics

3. Khanlari, A. and Mansourkiaie, F. (2015). Using robotics for STEM education in primary/elementary schools: Teachers' perceptions. 2015 10th International Conference on Computer Science and Education (ICCSE). [último acceso: 05/06/2016] Disponible en: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7250208&newsearch=true&queryText=education %20robotics

4. Wilson, S., Gameros, R., Sheely, M., Lin, M., Dover, K., Gevorkyan, R., Haberland, M., Bertozzi, A. and Berman, S. (2016). Pheeno, A Versatile Swarm Robotic Research and Education Platform. IEEE Robot. Autom. Lett., 1(2), pp.884-891. [último acceso: 05/06/2016] Disponible en: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7397909&newsearch=true&queryText=education %20robotics

5. R. Mitnik, M. Nussbaum, and A. Soto, "An autonomous educational mobile robot mediator,"AutonomousRobots,vol. 25, no. 4, pp. 367– 382, Nov. 2008.[último acceso 05/06/2016] Disponible en: http://dx.doi.org/10.1007/s10514-008-9101-z

6. F. R. Sullivan, "Robotics and science literacy: thinking skills, science process skills and systems understanding,"Journal of Research in Science Teaching,vol. 45, no. 3, pp. 373–394, Mar. 2008.[último acceso: 05/06/2016] Disponible en: http://dx.doi.org/10.1002/tea.20238

7. M. Barak and Y. Zadok, "Robotics projects and learning concepts in science, technology and problem solving,"International Journal of Technology and Design Education,vol. 19, no. 3, pp. 289–307, Aug. 2009. [último acceso: 05/06/2016] Disponible en: http://dx.doi.org/10.1007/s10798-007-9043-3

8. M. Rocard, et al., "Science education now: A renewed pedagogy for the future of Europe," Luxembourg, Office for Official Publications of the European Communities. [último acceso: 05/06/2016] Disponible en: http://ec.europa.eu/research/science-society/document_library/pdf_06/report-rocard-on-science-education_en.pdf

9. Fang, Z., Fu, Y. and Chai, T. (2009). A low-cost modular robot for research and education of control systems, mechatronics and robotics. 2009 4th IEEE Conference on Industrial Electronics and Applications. [último acceso: 05/06/2016] Disponible en:

http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5138725&newsearch=true&queryText=education%20robotics

10. ABC. (2016). Sacarino a su servicio - ABC.es. [online] Disponible en: http://www.abc.es/20110306/comunidad-castillaleon/abcp-sacarino-servicio-20110306.html [Último acceso: 11/09/2016].

11. Wolf, M. and McQuitty, S. (2011). Understanding the do-it-yourself consumer: DIY motivations and outcomes. AMS Review, 1(3-4), pp.154-170.

Análisis Visual Interactivo de Espacio-Tiempo Narrativo

Eduardo Flores González[1] y Roberto Therón Sánchez[1]

Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n 37008 Salamanca, Espańa
cradud@usal.es, theron@usal.es

Resumen En cualquier narración en la que intervienen varios personajes en un espacio-tiempo narrativo puede ser muy complejo llegar a comprender todas las interacciones que tienen lugar. Las herramientas visuales que son capaces de computar grandes volúmenes de información para convertirlos es formas de representación interactivas que faciliten esta comprensión son de inestimable ayuda. El objetivo perseguido es la investigación en procesos y técnicas que son capaces de extraer y representar las relaciones complejas que están implícitas en toda narración.

Keywords: Visualización de la Información, Analítica Visual, Guiones Cinematográficos, Análisis Estructura, Minería de Opinión, Análisis de Sentimientos

1. Introducción

Toda película tiene su punto de partida en la creación de un guión previo donde se estructuran sus escenas, acciones, personajes y diálogos. La industria cinematográfica genera una gran cantidad de documentos de este tipo cada ańo. Por otro lado, estos guiones, además intentan reflejar de alguna manera otros aspectos no estructurales, como pueden ser sentimientos y motivaciones que buscan los personajes a lo largo de la película, mediante acciones y diálogos.

El presente trabajo propone, por tanto, además de una representación visual de guiones de cine, cuyo objetivo es proporcionar una visualización eficaz capaz de transmitir de manera adecuada los elementos más relevantes del guión, un análisis de los sentimientos que transmiten los diferentes personajes a través de sus diálogos y la secuencia de escenas.

2. Guiones cinematográficos

Un guión cinematográfico es un documento de producción en el que se especifican los detalles necesarios para la realización de una película. Su estructura básica contiene escenas, acciones, diálogos entre personajes, descripciones del espacio donde se desarrolla, e incluso puede contener acotaciones breves para los actores, indicando qué emoción deben trasmitir a la hora de interpretar. Es por

ello que un guión debe aportar la suficiente información para que el lector se haga una idea general de la película en su mente.

A lo largo del tiempo, la estructura de los distintos guiones ha tendido a converger, facilitando así el trabajo de los profesionales del mundo cinematográfico, aplicándose dicha estructura común.

Calvisi [1] propone en su libro una estructura de guión denominada Story Map. Según él, el 95 % de las películas siguen esta estructura. Además, el autor eleva el porcentaje al 100 % cuando se tienen en cuenta las películas comerciales de Hollywood. Por esta razón se sigue esta estructura a la hora de evaluar la representación y captura de sentimientos propuesta.

3. Análisis de sentimientos

El análisis del sentimiento es el proceso por el que es posible determinar si una frase o acto de habla contiene una opinión, positiva o negativa, sobre una entidad concreta o sobre un concepto. Actualmente, es un término que está muy ligado a las redes sociales pero que, en realidad, no está limitado a ellas.

Mediante el análisis del sentimientos, se tiene como objetivo entender, en primer lugar, con qué guarda relación el texto que se analiza. Una vez que el sistema tiene claro sobre qué es la opinión, el segundo objetivo es conocer el sentido de esa opinión (básicamente positiva o negativa). A esa puntuación, técnicamente se la conoce como «intensidad de la polaridad».

Los sistemas más simples se limitan a leer una frase y buscar en ella palabras que tienen registradas en su diccionario como positivas o negativas. Si aparece una palabra positiva es una opinión positiva, y de manera opuesta para el caso de una palabra que sea considerada como negativa. En el caso de no encontrar ninguna palabra, que según el diccionario exprese algo positivo o negativo, se suele hablar de «opiniones neutras», aunque sería más correcto decir que lo que ocurre es que no hay opinión. Otros sistemas también clasifican una opinión como neutra si han encontrado una palabra positiva y otra negativa, cosa que no tiene por qué ser válida.

Dos complementos que son de ayudar para llevar a cabo las prácticas de análisis de sentimiento comentadas son:

- Un motor de monitorización (también conocido como «*crawler*»), que encuentra los textos que se deben analizar.
- Una herramienta de visualización y explotación de la información, que recoge la información proporcionada por el motor de análisis del sentimiento y compone con ella la interfaz de usuario.

3.1. Estado del arte en la Minería de opinión

La calificación sentimental es un objeto amplio de estudio actualmente y muchos investigadores del ámbito de la recuperación de información y la lingüística computacional han llevado sus investigaciones a esta rama. Por ejemplo Hatzivassiloglous y McKeown trabajaron en la orientación semántica (polaridad) de

adjetivos [2]. Desde entonces, se han empleado varias técnicas en la minería de opiniones.

No obstante se puede considerar que, verdaderamente, los primeros trabajos para analizar y clasificar la polaridad de un texto datan a Turney [4] y Pang [3] que aplicaron diferentes métodos para detectar la polaridad de críticas de productos y películas respectivamente.

También es posible clasificar la polaridad de un documento en una escala de varios valores, lo cual fue intentado por el mismo Pang [5] y Snyder [6], el cual realizó un análisis en profundidad de críticas a restaurantes, prediciendo evaluaciones par varios aspectos del restaurante dado, tales como la comida y ambiente del establecimiento en una escala de 5 estrellas. Además de Vryniotis [7], extendiendo la tarea de clasificar una crítica de película como positiva o negativa a predecir evaluaciones en una escala de 3 ó 4 estrellas.

A pesar de que en la mayoría de métodos de clasificación estadísticos la clase neutral es ignorada bajo la suposición de que los textos neutrales son binarios, varios investigadores sugieren que, al igual que en todo problema de polaridad, han de ser identificadas 3 categorías.

Además, ha sido probado que algunos clasificadores específicos tales como el de Máxima Entropía del citado Vryniotis y las Máquinas de vectores de soporte con Koppel y Schler [8] se benefician de tener una clase neutral, mejorando la precisión de la clasificación.

Un método diferente para determinar sentimiento es el uso de un sistema de escalado donde a las palabras que normalmente se asocian con un sentimiento negativo, neutral o positivo se les asigna una escala (por ejemplo de -10 a 10). A cada concepto se le otorga una puntuación basada en la forma en que las palabras asociadas con sentimientos se relacionan con el concepto y su puntuación. De forma alternativa se puede otorgar a los textos una puntuación por la intensidad de sentimientos positivos y negativos si el objetivo es determinar el sentimiento en un texto en lugar de la polaridad e intensidad general del texto [9].

Otra dirección de investigación es la identificación subjetivo / objetivo. En este sentido Pang mostró que eliminar las oraciones objetivas de un documento antes de clasificar su polaridad ayudó a mejorar su rendimiento [10].

Un modelo de análisis más detallista es el llamado análisis de sentimientos basado en rasgos / características descrito por Hu y Liu [11]. Se refiere a determinar opiniones o sentimientos expresados sobre diferentes rasgos o características de entidades. Este problema engloba a su vez varios problemas más concretos como identificar entidades relevantes, extraer los rasgos/características y determinar si una opinión expresada sobre cada rasgo/característica es positiva, negativa o neutral [12].

Más recientemente, se pueden destacar varios trabajos importantes sobre ontologías y minería de opiniones basadas en características. El objetivo de estos trabajos es calcular la polaridad teniendo en cuenta las características de un concepto para una ontología.

A diferencia de Hu y Liu [13], en esos trabajos se utilizan ontologías del dominio que modelan el contenido de textos en el corpus y, a partir de dichas ontologías, extraen los adjetivos que describen dichas características. En el tra-

bajo de Zhou y Chaovalit, la polaridad de un texto se calcula basándose en los pesos de las características, y el valor de la polaridad de la característica se calcula mediante la «estimación de máxima verosimilitud».

En el trabajo de Zhao y Li, se obtiene la orientación semántica del texto desde la jerarquía de la ontología. Además, el método que proponen es capaz de obtener la posibilidad, negatividad y neutralidad de un texto. Estos dos trabajos demuestran que el uso de ontologías mejora los resultados en el problema del análisis sentimental.

3.2. Soluciones para el análisis de sentimientos

Muchos de los algoritmos de análisis de sentimientos basados en una clasificación usan una colección de datos. Antes de aplicarles el analizador, los datos son preprocesados para extraer las características principales. Son varios métodos de clasificación los que han sido propuestos: Naive Bayes, Suport Vector Machines (SVM), KNN, etc. No obstante, de acuerdo con Richa Bhayani y Lei Huang [14] no está claro cual de estas estrategias de clasificación es la más apropiada para realizar análisis de sentimientos.

En este trabajo, se hace un primer acercamiento del análisis de sentimiento para guiones cinematográficos y su representación visual, por lo que se centra en aplicar un método de análisis sentimental bayesiano para un contexto concreto como es el de los guiones cinematográficos, de manera que se pueda expresar la polaridad de las escenas de forma independiente y la del conjunto de sus personajes para evaluar si en una categorización temática de películas se siguen patrones "sentimentales"dentro de las mismas, así como la evolución de los personajes a lo largo del guion.

En este trabajo se ha decidido hacer uso de una estrategia de clasificación basada de Naive Bayes porque es un método simple e intuitivo cuyo rendimiento es similar al de otros enfoques. Además, este clasificador ha sido usado satisfactoriamente en la web, desde clasificaciones de IMDB hasta filtros de spam.

4. Estado del arte en la Visualización de la información

La Visualización de la Información es un concepto complejo, un proceso del cual se han propuesto distintas definiciones en sentidos diversos. Por ejemplo, los autores Card, Mackinlay y Shneiderman [15] afirman que la Visualización de la Información es el uso de representaciones visuales interactivas soportadas por ordenador, de datos abstractos para amplificar la cognición.

Otros, como Colin Ware [16], hablan de que se trata de una actividad cognitiva, algo que ocurre en la mente y que tiene como resultado algo relativamente efímero como son los pensamientos, algo que no puede ser impreso en papel o visto en un microscopio.

El tercer tipo de acepción para este concepto, lo da Spence [17], el cual afirma que se trata de un modelo mental, un modelo interno o un mapa cognitivo, como construcciones mentales disponibles para una inspección mental.

En cualquier caso, en lo que coinciden todos los autores, es que la Visualización de Información es un proceso por el cual el usuario es capaz de llegar al conocimiento y la sabiduría a través de la observación de representaciones visuales de datos o información.

4.1. Soluciones para la visualización de datos

Este proceso consiste en pasar de los datos iniciales, que tienen una estructura potencialmente desordenada y sin lógica, a una representación visual final que represente dichos datos al observador final. Además, en este proceso intervienen las acciones de manipulación, interacción y transformación por parte del observador, que afectan directamente la representación final generada.

5. Representación visual

5.1. Aspectos técnicos

Para la recopilación, procesado de sentimientos y generación de nuevos ficheros XML con los atributos de sentimientos se ha empleado el lenguaje de programación PHP, el cual es un lenguaje de licencia libre y gratuito para usarlo en cualquier tipo de desarrollo, incluso en aquellos con carácter comercial. La sencillez de sintaxis, la velocidad de ejecución y rendimiento, además su amplia comunidad de usuarios hace que la cantidad de recursos y de información disponible del mismo, justifica el uso del lenguaje. Como entorno de desarrollo para PHP se ha usado un simple editor de texto llamado Sublime Text, que permite realizar un desarrollo rápido gracias al resaltado de sintaxis y a que incorpora bastantes herramientas para la edición y maquetación del texto de forma sencilla. En cuanto al desarrollo de la interfaz y visualización propuestas en este trabajo fin de máster, se ha hecho uso de HTML5, CSS3 y Javascript. Además, se han utilizado varias librerías basadas en Javascript como son AngularJS, jQuery y D3JS. **D3JS** es una biblioteca Javascript open source que permite crear gráficos interactivos y dinámicos en navegadores web, muy usado para la implementación de gráficos complejos. Hace uso de SVG, HTML5 y CSS. En contraste a otros librerías usadas para la visualización de datos, ésta permite un control bastante bueno sobre la visa final resultante.

5.2. Extracción de datos

Las escenas se introducen con el lugar de la escena donde se desarrolla las mismas. A continuación, le sigue el lugar y finalmente información adicional (por ejemplo, si la escena transcurre durante el día o la noche). Después comienza la descripción de la escena, que suele incluir el escenario, actitud de los personajes, localizaciones o acciones entre otros. Tras la descripción, se puede ver los diálogos de los personajes que tienen también una estructura predefinida.

5.3. Ficheros de guión en formato XML

Un fichero XML permite identificar los elementos disponibles en la representación. Sin embargo, no todos los datos contenidos en este fichero son interesantes para la representación ya que pueden aportar información sin interés o sobrecargar la visualización al observador si se incluyesen. En este trabajo, no obstante, se aprovecha gran parte de la información contenida en el guion. Posteriormente se definen los atributos seleccionados para incluir en la representación (personajes y escenas) y las técnicas que se utilizan para «mapear» los atributos en elementos visuales. El siguiente punto trata sobre la generación de sentimientos, es por ello, que es necesario comentar que posterior al procesamiento del fichero XML con la estructura del guión, se genera un nuevo fichero XML con los nuevos atributos generados después de haber realizado el análisis. Este fichero XML, es el que se utilizará finalmente para la representación visual.

5.4. Generación de sentimientos

Por un lado se obtiene la información de los diálogos de los personajes para hacer un análisis de los mismos. Para ello, dentro de las escenas, definidas en el fichero XML con la etiqueta «timeSlice», se buscan los distintos diálogos que forman parte de la escena. Puesto que una escena está compuesta de una o varias acciones, hay que buscar todas las acciones (etiqueta «accion») de cada escena y dentro de la misma todos los elementos de diálogo. Éstos se encuentran bajo la etiqueta «charDialog». Por último, el diálogo se encuentra dentro del atributo «texto» y corresponde al personaje del atributo «char». Por otro lado las acciones que conforman las distintas escenas de la película, se encuentran dentro del atributo «descripción» de la ya comentada etiqueta «acción». Una vez que son analizados los datos partiendo de esos parámetros de entrada, los datos devueltos ayudan, utilizando un sistema simple de asignación de sentimiento para elementos como personajes y escenas, a añadir atributos de sentimiento a estos nuevos elementos, además de los analizados. Los elementos a los que se le añade atributos de sentimiento son los siguientes:

- **Listado de personajes:** cada personaje cuenta con un atributo que no es más que la media global del personaje durante la película.
- **Escenas:** cada escena cuenta con una valoración de sentimiento que viene dada por la media de los sentimientos de personajes y acciones que conforman la escena.
- **Personajes que forman la escena:** cada personaje cuenta con una valoración sentimental acorde con el diálogo que mantenga en la escena.
- **Acciones:** cada acción cuenta con una valoración sentimental arrojada por el analizador.
- **Diálogos:** cada diálogo de las escenas es analizado individualmente para mostrar el resultado de sentimiento obtenido.

Cabe destacar que se ha incluido una nueva categoría «neutral» para aquellos casos en los que a través de el cálculo posterior de sentimiento en las que intervienen varios elementos.

Clasificador Naive Bayes Para realizar la minería de sentimientos se ha utilizado un sistema basado en el clasificador Naive Bayes. Se trata de una técnica de clasificación y predicción supervisada que construye modelos que predicen la probabilidad de posibles resultados. Constituye una técnica supervisada porque necesita tener ejemplos clasificados para que funcione.

Para el conjunto de datos se ha hecho uso de un diccionario de 10000 frases perteneciente a opiniones de cine. Los datos se suministran en dos archivos, uno con 5000 frases con frases categorizadas como positivas y otro para las negativas. Este diccionario fue por primera vez usado en [5]. Cada línea de los dos ficheros corresponden a una frase y según en que fichero estén, estarán ya clasificadas como positivas o negativas.

5.5. Tratamiento de datos

El eje central del proyecto, que es el análisis de sentimientos tiene como elementos de partida para la implementación de la herramienta de análisis de sentimientos las escenas y los personajes. De esta manera se considera a los sentimientos de los elementos anteriores como atributos. Las técnicas y procesado de datos en el proceso de visualización utilizados son los siguientes.

Variables Los datos fundamentales en un guión y en su correspondiente estructura giran en torno a los personajes y la secuencia de escenas del guión. Relativo a los sentimientos, es una información subjetiva y que no es posible sacar a partir de los datos implícitos del guión. Para el resultado del análisis de sentimiento se ha utilizado una polaridad binominal para diferenciar entre positivo y negativo. De aquí se añaden atributos a las diferentes variables.

Personajes Los elementos más importantes relacionados con los personajes son:

- Nombre: identifica a éste a lo largo del guion.
- Importancia del personaje: se define como el número de veces que éste aparece en las escenas del guión.
- Evolución del personaje: el orden y momentos en los que un personaje va apareciendo por las distintas escenas en las que interviene.
- Relaciones entre personajes: aquellas escenas en las que aparecen simultáneamente varios personajes puede significar una interacción entre los mismos.
- Sentimiento: el sentimiento es un elemento abstracto, pero que es realmente importante en el desarrollo de una película. Es interesante valorar la evolución sentimental de un personaje para valorar si el desarrollo de la historia influye en lo que transmite.

Escenas Las escenas son eventos que tienen lugar enteramente en una localización o tiempo concretos. La información fundamental de las escenas que se consideró relevante para la representación es:

- Número: se incluye el número de escena que ocupa dentro del guion para identificarla.

- Datos de cabecera: incluye localización, y la información relativa a si la escena transcurre en interior o exterior.
- Número de personajes que aparecen: el número de personajes que aparecen marca en muchos casos la importancia de una escena en la que convergen varias historias paralelas.
- Duración de la escena: el análisis de esta información ayuda a discriminar entre aquellas escenas rápidas de unos segundos de duración y otras más largas y que potencialmente tengan más importancia para la historia.
- Acciones: las acciones describen qué es lo que sucede en la escena, lo que sirve para dar coherencia al resto de información asociada a la misma.
- Diálogo: los diálogos son la parte más detallada de una escena, ya que justifican en cierta medida lo que una acción describe.
- Sentimiento: El sentimiento en una escena es un factor complejo de medir. El sentimiento en una escena está compuesto por el sentimiento parcial que transmiten las acciones que forma la escena, así como el sentimiento de cada uno de los diálogos que se dan en la misma.

Atributos de personaje Una vez establecidos los elementos que van a representarse en la visualización, se especifican aquellos atributos que van a formar parte de los elementos visuales que conforman la representación visual.

Personajes Los personajes al ser elementos categóricos se utiliza el color para representarlos. La elección de colores adecuados para la representación no es trivial e influye directamente en la eficacia de la representación para transmitir información. En este caso, la elección de colores está basada en los trabajos de Brewer et al. [18] y Bostock et al. [19] donde se muestran qué colores resultan adecuados para la representación de atributos categóricos y que colores minimizan las dificultades de identificación para personas con problemas de percepción de los colores. Además, los colores seleccionados mejoran notablemente la estética final de la representación con respecto a otros esquemas de colores planteados.

Nombres de Personajes Los nombres de los personajes, se consideran como atributo de texto, por lo que se usan etiquetas que muestran el texto correspondiente cuando es requerido por el analizador. Cuando el cursor del ratón se sitúa sobre la línea de un personaje, se muestra una etiqueta del personaje.

Importancia de Personajes Se considera la importancia de un personaje como el número de escenas en las que aparece en la película. Dicha importancia se muestra visualmente a través del ancho de la línea de cada personaje. Esta forma de representación es adecuado para el proceso perceptivo, permitiendo la rápida identificación de los personajes más importantes.

Evolución espacio-temporal de Personajes La evolución de los personajes se contempla en función de dos variables, estas son, espacio y tiempo. La variable de espacio se define como las secuencias en las que interviene cada uno de ellos.

La variable temporal responde a la interacción que cada personaje tiene con la evolución de cada una de las escenas ordenadas por orden de aparición, mostrándose visualmente mediante una línea temporal que recorre horizontalmente cada una de las escenas de la película desde que el personaje hace su primera aparición hasta que finaliza la interpretación en su última escena.

Relaciones entre personajes Las interacciones que se tienen entre los personajes que conforman el guión, se puede observar a través de las propias escenas donde convergen personajes. A través de esta representación se puede encontrar patrones muy interesantes.

Sentimiento de personajes El sentimiento de cada uno de los personajes se contempla en función del sentimiento global que cada personaje tiene a lo largo del guión. Dicho sentimiento se calcula a través de cada uno de los sentimientos parciales que el personaje tiene en las escenas en las que interviene. Para calcular dicho sentimiento se ha definido la siguiente fórmula:

$$S_{personaje} = \frac{\sum_{i=1}^{n} s_1 + s_2 + ... + s_n}{N_{personaje}}$$

El resultado dará un valor comprendido entre -1 y 1 considerándose como sentimiento global positivo si $S_{personaje}$ está comprendido entre $(0, 1]$, neutro si es 0 y negativo si el valor está comprendido entre $[-1, 0)$.

La característica elegida para mapear el sentimiento de los personajes en la representación visual es el color, ya que es una característica adecuada para mapear atributos de sentimiento según se indica en Borth et. al [20]. Dicho color es reflejado en el borde del nombre del personaje, lo que permite de forma ágil identificar el sentimiento de todos los personajes que conforman el guion. Los colores que se utilizan para la representación siguen la idea de definir los sentimientos a través de los colores primarios: verde (positivo), rojo (negativo) y azul (neutro).

Atributos de escena Las escenas corresponden a las variables espacio-temporal de un guion ya que ocurren a lo largo de momentos puntuales de tiempo y en ubicaciones determinadas. Por esta razón, las escenas se posicionan secuencialmente a través del eje horizontal de la representación. Las escenas contienen mucha información que es útil a la hora de analizar un guión, como son los personajes. Es por ello que la forma elíptica que se da a la escena permite representar los elementos contenidos en la escena. Otros elementos importantes que aportan información a la escena, son los siguientes:

Número de escena Se trata de un valor numérico que va desde el 1 hasta el número total de escenas. Debido a la representación horizontal a través de la que se van representando de manera secuencial cada una de las escenas, los valores numéricos se pueden representar en las secciones verticales que contienen la escena asociada a cada número de escena, ya que dichos números no se solaparán

y se podrán asociar fácilmente a las escenas correspondientes. Además, se añade la opción de dibujar unas líneas verticales que apoyan la asociación perceptiva de los números con las escenas correspondientes.

Datos de cabecera Son aquellos datos relativos a la escena como el espacio en el que se desarrollan, además de información si la escena transcurre en un escenario interior o exterior. Esto se muestra a través de iconos, ofreciéndose así información muy intuitiva sobre la escena, facilitando el proceso de análisis.

Número de personajes en escena A través de los puntos de personaje, se puede ver de forma rápida el número de personajes que tienen lugar en una escena. Además, cada uno de los puntos tiene el color asociado al personaje que interviene en esa escena, por lo que es muy rápido ver si una escena la conforma personajes que no tienen un gran peso en la historia o, en cambio, si en esa escena interviene el protagonista, antagonista o incluso ambos. También el número de personajes en escena permite medir su potencial importancia que tiene en la película.

Personajes en escena Es el listado de personajes que conforman la escena y que se muestran en el detalle de la misma. Dichos personajes se muestran pintados en el color que representa la línea de personaje correspondiente al mismo, lo que facilita la asociación de personajes entre la vista general del guión y el detalle de la escena a la persona que analiza el mismo.

Además del nombre del personaje y el color correspondiente, cada nombre está rodeado por un rectángulo que refleja el sentimiento del personaje dentro de la escena. Dicho sentimiento está formado por la suma de los sentimientos que transmite el personaje a lo largo de los diálogos que se dan en la escena. Viene dado por la siguiente fórmula:

$$S_{personaje} = \frac{\sum_{i=1}^{n} Sd_1 + Sd_2 + ... + Sd_n}{Nd_{personaje}}$$

La característica elegida para mapear el sentimiento de los personajes en la representación visual es el color reflejado en el borde del nombre del personaje, lo que permite de forma ágil identificar el sentimiento de todos los personajes que conforman el guion. Los colores que se utilizan para la representación siguen la idea de definir los sentimientos a través de los colores primarios.

Acciones en escena Una escena está formada por una o varias acciones generalmente, aunque puede darse el caso en el que una escena no contenga ninguna acción. Las acciones son atributos textuales que describen detalladamente lo que tiene lugar en la escena.

El texto se muestra de forma completa en la visualización, dentro del detalle de escena, lo que ofrece de un vistazo al guionista el número de acciones que hay y el orden en el cual que aparecen en el guión. Además, el guionista puede realizar un análisis más a bajo nivel y de forma somera, de dichas acciones.

Al igual que en listado de personajes que intervienen en la escena, en las acciones también se obtiene un valor de sentimiento que viene dado por un análisis de sentimiento utilizando un modelo bayesiano, a través de métodos supervisados. La forma de representar el sentimiento es a través del color, el cual es reflejado en el borde de cada acción de la escena, lo que permite de forma ágil identificar el sentimiento de todas las acciones. Los colores que se han elegido para representar los sentimientos de las acciones, son nuevamente los colores primarios.

Diálogos en escena Un diálogo es un atributo textual y se puede considerar como el elemento a más bajo nivel que se da lugar en la escena. Los diálogos están clasificados dentro de las acciones y al igual que ocurre con las acciones, puede haber casos en los que exista una acción en la escena, pero no exista ningún diálogo.

Cada diálogo se representa dentro de un rectángulo, y es pintado secuencialmente en el orden en el que aparece en el guión. A su vez, todos los diálogos que pertenecen a una acción se encuentran dentro del rectángulo que contiene la propia acción. De esta manera, el guionista que analizará el documento, le resultará bastante sencillo distinguir la importancia de una escena e incluso de una acción, por el número de diálogos que contenga el mismo. También, de cada diálogo se ańade el nombre del personaje, lo que permite reflejar visualmente la importancia de un personaje dentro de la misma escena.

Sentimientos en escena Capturar el sentimiento de una escena es algo complejo ya que intervienen 3 elementos para obtener el valor del mismo. Por un lado se encuentran las acciones que conforman la escena, por otro lado los personajes que intervienen en la escena y por último los diálogos que se dan en cada una de las acciones de la escena. Para obtener el valor del sentimiento de la escena se utiliza la siguiente fórmula:

$$S_{escena} = \frac{\sum_{i=1}^{n} Sa_1 + Sa_2 + ... + Sa_n + \sum_{i=1}^{n} Sd_1 + Sd_2 + ... + Sd_n + \sum_{i=1}^{n} Sp_1 + Sp_2 + ... + Sp_n}{N_{acciones} + N_{dialogos} + N_{personajes}}$$

La característica elegida para mapear el sentimiento de los personajes en la representación visual es el color, como en los elementos anteriores. Dicho color es reflejado en el borde de la escena, lo que permite de forma ágil identificar el sentimiento de la escena en el caso de que se esté previsualizando el detalle de escena, o de todas las escenas, si se está observando la primera visualización.

Características de diseńo A lo largo del desarrollo del presente trabajo se ha decidido seguir unas pautas de diseńo que confieren a la representación. Estas características, junto con las tomadas como base de los trabajos antecedentes a éste, proporcionan una herramienta con un potencial importante en la representación de guiones y que tiene como objetivo principal servir como herramienta didáctica y ayuda en la escritura y posterior análisis de un guion de cine en el

que el guionista sabe lo que quiere conseguir, pero en los muchos cambios que hará desde que empiece hasta que termine, la visualización le ayuda a ver cómo va quedando tanto la estructura del documento cómo a nivel de sentimientos.

Transiciones Existen varias formas de enfocar aquellas situaciones en las que se producen cambios en la imagen final generada como cuando existe un cambio en los datos que se muestran. Uno de esos enfoques puede ser un cambio estático, donde una representación desaparece, generándose de forma inmediata una nueva con los nuevos datos. Otro enfoque más óptimo, es realizar animaciones que sirvan de transición entre la primera imagen y la última lo que permite mantener al usuario en contexto durante los cambios de los datos. Algunos estudios como el de Heer and Robertson [21], demuestra una mejora importante en la percepción gráfica en los cambios entre distintos gráficos de datos, por lo que el uso de transiciones es una técnica casi imprescindible a utilizar para mejorar la capacidad comunicativa de una representación. En este sentido, se aplican en la medida de lo posible transacciones en varias interacciones que provocan cambios en los datos.

Interacciones Las interacciones son elementos muy importante en una representación visual, por lo que se han introducido una serie de interacciones que ayudan a una mejor comprensión de la información disponible. Estas interacciones se realizan sobre los personajes, las líneas de personajes, las escenas y otras globales que afectan al conjunto de la visualización.

6. Caso de estudio

Se ha hecho un breve análisis un guion usando la representación utilizada y se estudian algunas de las características que se pueden extraer haciendo uso de la representación de sentimientos, haciendo una comparación con el documento de Story Boards de Calvisi [1].

6.1. Rocky

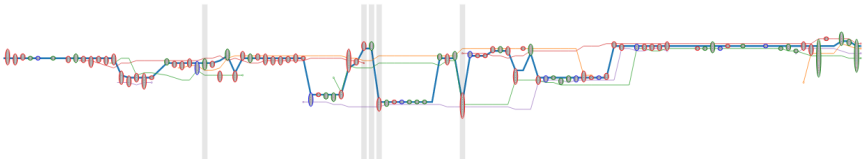


Figura 1: Esta película escrita por Sylvester Stallone y dirigida por John G. Avildsen en 1976 con una duración de 119 minutos, tiene varias características estructurales. Esta imagen arroja una mayoría de puntos rojos en las escenas, lo que se traduce como que es una película en la que predomina el sentimiento negativo.

6.2. Acto 1

10 - Inciting Incident (Internal) : *Rocky visits the pet store to see shy Adrian. She doesn´t respond to his jokes.*

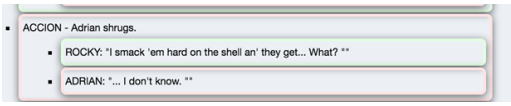


Figura 2: Esta es la primera escena donde Rocky se encuentra con Adrian. Él hace bromas para intentar interactuar con ella, pero ella, quizá por timidez, no muestra interés. El análisis visual que muestra la imagen, indica que en el diálogo de Adrian hay un sentimiento negativo y la acción que simplemente indica que Adrian se encoge de hombros, coincide también el sentimiento negativo. En cambio Rocky, en ese diálogo en el que intenta interactuar con ella, el sentimiento es positivo.

6.3. Acto 2A

40 - First Trial/First Casualty : *Rocky and Adrian's date at the ice rink. They make a connection when Rocky says his father told him he had a body and no brain, and Adrian says her mother said she didn't have a body so she should develop her brain.*

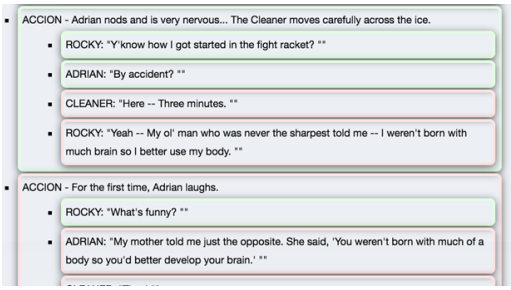


Figura 3: La visualización marca esta escena como negativa debido principalmente a la carga sentimental de la misma, ya que Rocky le explica a Adrian que cuando era pequeño su padre le dijo que no tenía cerebro y Adrian confiesa a Rocky que en su niñez su madre le dijo que era una chica fea y que tendría que desarrollar su cerebro. De hecho los diálogos en los que los personajes cuentan esto se marcan como negativos.

6.4. Acto 2B

62 - 69 : *Mickey asks Rocky if he can be his manager and they argue.*

Esta parte, mostrada en la figura 4, tiene una gran carga emocional, ya que aquí Mickey, el antiguo entrenador de Rocky, después de haberle quitado la taquilla en el gimnasio pide a Rocky ser su entrenador en el combate contra Apollo. La visualización marca los diálogos de Mickey en positivo y los de Rocky en negativo, esto es coherente ya que Mickey quiere convencer a Rocky y éste se muestra dolido.

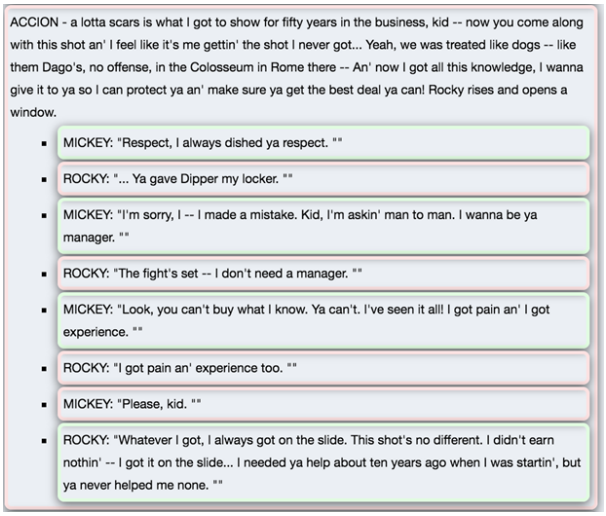


Figura 4: Rocky - Detalle de escena 50

6.5. Acto 3

97 - : *Rocky admits to Adrian he can't beat Apollo Creed.*

En la figura 5 Rocky admite a Adrian que no va a poder vencer a Apollo. La visualización marca casi todos los diálogos de Rocky como negativos. En cambio las acciones están marcadas como positivas, ya que en la descripción de las mismas no se refleja la intensidad de los diálogos.

7. Conclusiones y Trabajo Futuro

Figura 5: Rocky - Detalle de escena 88A

En este trabajo se ha desarrollado una representación visual de guiones cinematográficas, con lo cual se ha cumplido el objetivo que se planteaba inicialmente; además se ha llevado a cabo teniendo en cuenta conceptos que forman parte de la revisión del estado del arte en la Visualización de la Información, con lo cual se han seguido una serie de patrones y guías para la creación de visualizaciones de datos adecuadas. Con lo cual se puede asumir que habiendo aplicado estas reglas y principios, la visualización propuesta tiene al menos cierta coherencia. Aún así, no se puede asegurar que esta visualización sea totalmente válida, ya que no ha sido probada por más que un conjunto mínimo de películas.

En este sentido, puede ser útil como línea de trabajo futura en cuanto a la visualización estructural del guión, una representación temporal de la historia, en el cual las escenas se presentarían por su orden cronológico, pudiendo así analizar estructuras como *flashbacks* y otros saltos temporales.

Por otra parte, en relación al apartado de análisis de sentimientos, si el conjunto de películas fuese más amplio y se aplicaran otro tipo de métodos de clasificación como Máquinas de Vectores de Soporte, podría desarrollarse un análisis más profundo del análisis de sentimientos en guiones de cine, de modo que las tareas de análisis que soporta puedan ser arropadas por otro conjunto de propuestas y funcionalidades de análisis.

Referencias

1. Daniel P. Calvisi. *STORY MAPS: How to Write a GREAT Screenplay* in Act Four Screenplays, 2012.
2. Vasileios Hatzivassiloglou and Kathleen R. McKeown *Predicting the Semantic Orientation of Adjectives* in Department of Computer Science Columbia University, 1997.
3. Pang, B.; Lee, L. y Vaithyanathan, S. *Thumbs up? Sentiment classification using machine learning techniques* in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 79-86, 2002.
4. Peter D. Turney *Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews* in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 417-424.
5. Pang, B.; Lee, L. *Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales* in 2005.
6. Benjamin Snyder and Regina Barzilay *Multiple Aspect Ranking using the Good Grief Algorithm* in Massachusetts Institute of Technology, 2007.
7. Vasilis Vryniotis *The importance of Neutral Class in Sentiment Analysis* in 2013 URL http://blog.datumbox.com/the-importance-of-neutral-class-in-sentiment-analysis/.
8. Moshe Koppel and Jonathan Schler *The Importance of Neutral Examples for Learning Sentiment* in Bar-Ilan University, 2006.
9. Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai *Sentiment Strength Detection in Short Informal Text* in University of Wolverhampton, 2010.
10. Bo Pang and Lillian Lee *A Sentimental Education: Sentiment Analysis Using Subjectivity* in 2004.
11. Hu, M. y Liu, B. *Mining and summarizing customer reviews* in ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 168-177, 2004.
12. Liu, B.; Hu, M. y Cheng, J. *Opinion observer: analyzing and comparing opinions on the web* in International Conference on World Wide Web, pages 342-351, 2005.
13. Lina Zhou and Pimwadee Chaovalit *Ontology-supported polarity mining* in Journal of the American Society for Information Science and Technology Volume 59, Issue 1, pages 98–110, 1 January 2008.
14. Richa Bhayani y Lei Huang *Twitter sentiment classication using distant supervision* in 2009.
15. Stuart K Card, Jock D Mackinlay, and Ben Shneiderman *Readings in information visualization: using vision to think* in information visualization: using vision to think. Morgan Kaufmann Pub, 1999.
16. Ware Colin *Information visualization: perception for design* in Morgan Kaufmann, 5(6):8, 2004.
17. Robert Spence and Mark Apperley *Data base navigation: an office environment for the professional* in Behaviour and Information Technology, 1982.
18. C. Brewer, M. Harrower and D. Heyman *Colorbrewer 2.0: color advice for cartography* in 2009. URL http://colorbrewer2.org/.
19. Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer *D3: Data-driven documents* in IEEE Trans. Visualization and Comp. Graphics (Proc. InfoVis), 2011.
20. Damian Borth, Rongrong Ji, Tao Chen, Thomas Breuel and Shih-Fu Chang *Large-scale Visual Sentiment Ontology and Detectors Using Adjective Noun Pairs* in ACM Multimedia Conference, Barcelona, Oct 2013.

21. Jeffrey Heer and George G Robertson *Animated transitions in statistical data graphics* in statistical data graphics. Visualization and Computer Graphics, IEEE Transactions on, 2007.

Técnicas de reconocimiento facial basado en partes para una mayor resistencia a la oclusión

Daniel López Sánchez[1], Angélica González Arrieta[1]

Departamento de Informática y Automática, Universidad de Salamanca.
Plaza de la Merced s/n. 37008, Salamanca, España
{lope,angelica}@usal.es

Resumen Este trabajo se centra en el diseño de técnicas de reconocimiento facial resistentes a la oclusión y de bajo coste computacional. Se estudia la modificación de los algoritmos clásicos para detectar y descartar las partes de las imágenes faciales que se encuentren ocluidas y se propone la integración de estos métodos con técnicas de reducción de dimensionalidad que permitan reducir el coste computacional de la identificación. Finalmente se presentan resultados experimentales que comparan el método propuesto con las alternativas clásicas y validan su eficacia.

Keywords: Reconocimiento facial, oclusión parcial, reducción de dimensionalidad

1. Introducción

Este trabajo se ha centrado en el diseño e implementación de nuevos algoritmos para el reconocimiento facial, especialmente adaptados para lidiar con la oclusión parcial de las imágenes faciales. Además, se ha puesto énfasis en la búsqueda de un balance entre eficacia y eficiencia computacional, garantizando así que los algoritmos puedan ser utilizados en entornos computacionales con prestaciones limitadas. El modelo propuesto puede entrenarse satisfactoriamente incluso con pocas imágenes, y sin necesidad de conocimiento *a priori* de la naturaleza de la oclusión que encontrará en fase de test.

La sección 2 hace un repaso sobre las principales estrategias propuestas en la literatura para el problema del reconocimiento facial, haciendo énfasis en los estudios que tratan de solucionar el problema de la oclusión parcial. El grueso de la propuesta de este trabajo se describe a nivel teórico en la sección 3, donde además se explican los conceptos previos necesarios para su comprensión. En la sección 4 se presentan los resultados de un buen número de experimentos que comparan las técnicas clásicas de reconocimiento facial con la propuesta de este trabajo en la tarea de reconocimiento facial bajo oclusión parcial. Finalmente, en la sección 6 se presentan las conclusiones finales de este trabajo y se esbozan las lineas de trabajo futuro.

2. Estado del arte

En esta sección se repasan los principales enfoques que ha tenido en la literatura el problema de la oclusión facial.

El autor de [1] sostiene la hipótesis de que gran parte de la pérdida de precisión que sufren los sistemas de reconocimiento facial frente a la oclusión parcial se debe a los errores de alineamiento inducidos por la oclusión. En ese sentido, propone una técnica que busca minimizar la distancia entre cada muestra del set de entrenamiento y una nueva observación, probando un conjunto finito de posibles variaciones en el alineamiento. Sus tasas de acierto reportadas para la base de datos ARFace son las mejores hasta la fecha. El principal problema de este método es que dispara el coste computacional: la búsqueda de la alineación óptima fuerza a realizar cientos de comparaciones por cada muestra del set de entrenamiento.

Otros trabajos [2,3], siguen un enfoque en cierta forma similar al nuestro, al dividir la imagen en una serie de zonas de oclusión. Después buscan modelar estas zonas de oclusión local por medio del algoritmo PCA y de un *Self Organizing Map* (SOM) respectivamente.

La mayoría de sistemas propuestos, incluyen un paso previo a la identificación en el que se detectan las zonas de la imagen que están afectadas por la oclusión. Algunos autores utilizan parches de imágenes ocluidas y sin ocluir, etiquetadas manualmente, para entrenar clasificadores que aprenden a distinguir las zonas ocluidas de las no ocluidas [4]. Esta aproximación tiene la desventaja de necesitar imágenes que presenten oclusión en la fase de entrenamiento. Además, si la oclusión que el sistema tiene que soportar en producción no es la misma que la de las imágenes de entrenamiento la eficacia del sistema se verá afectada. Otros enfoques realizan segmentación basada en color [5], estos enfoques son muy sensibles a cambios en la iluminación, además asumen que la oclusión no puede ser causada por un objeto de color similar al de la piel humana.

Más recientemente, numerosos autores han tratado de aplicar los recientes avances en el campo del *deep learning* al problema del reconocimiento facial. Estos modelos aprovechan la desbordante cantidad de información y los dispositivos de computación masivamente paralela que se han vueltos disponibles en los últimos años para entrenar clasificadores robustos a condiciones no controladas gracias a su complejidad. Actualmente, el estado del arte en una de las bases de datos de reconocimiento facial más ampliamente utilizadas, *Labeled Faces in the Wild* (LFW), lo ostentan los científicos de la compañía china Baidu [6]. Esencialmente, los autores proponen una arquitectura en la que varias redes neuronales convolucionales profundas son entrenadas en paralelo sobre diferentes parches de la imagen original. La salidas de la última capa de convolución de cada red se concatenan para formar el descriptor final. Estas aproximaciones tienen su principal desventaja en el coste computacional y la cantidad de información que requieren para entrenarse.

3. Método propuesto

En esta sección se describe tanto el algoritmo de clasificación propuesto como las técnicas de preprocesamiento y diferentes métodos de extracción de características que serán utilizadas dentro de la sección de resultados experimentales. El procesamiento que sigue una imagen es el siguiente: (1) se detecta una región de interés para el rostro, (2) se alinea el rostro detectado, estimando una serie de puntos clave faciales, (3) se normaliza la pose, es decir, se genera una imagen normalizada en la que el rostro ha sido rotado y escalado a una configuración fija, (4) se normaliza la iluminación en la imagen, (5) se extraen las características y (6) se aplica algún algoritmo de clasificación, bien sea para su entrenamiento o para predicción. En particular el nuevo método de clasificación propuesto en este trabajo se describe en la sección 3.3.

3.1. Preprocesamiento

Este subapartado pretende proporcionar los detalles necesarios para reproducir el preprocesamiento aplicado en los experimentos como paso previo al reconocimiento facial.

Detección facial La tarea de los métodos de detección facial es encontrar una región de interés o *Region Of Interest* (ROI) aproximado para cada rostro humano que aparece en una imagen. El descriptor de imagen *Histogram of Oriented Gradients* (HOG) [7] contabiliza el número de ocurrencias de cada orientación de gradiente en regiones localizadas de una imagen. La idea fundamental de los detectores de objetos basados en HOG es aplicar un clasificador lineal a una ventana deslizante sobre el descriptor HOG de una imagen.

Durante los experimentos llevados a cabo en este trabajo, se ha hecho uso de la implementación de detector de objetos HOG proporcionada por la biblioteca Dlib C++ [8].

Alineación facial El proceso de alineación facial (o detección de puntos faciales característicos) consiste en predecir de forma automática la localización de una serie de puntos faciales clave en base a la imagen de un rostro humano y un ROI. Actualmente los modelos más empleados son los basados en regresión en cascada, gracias a su buen desempeño en comparación con las técnicas clásicas y su mayor rapidez. El modelo elegido para la alineación facial en este trabajo ha sido el propuesto por V. Kazemi en 2014 [9]. El autor propone entrenar un modelo de regresión en cascada de forma que en cada nivel un regresor se encargue de actualizar o desplazar los puntos faciales para acercarlos a su posición correcta. La figura 1 muestra el proceso de estimación y actualización de los puntos faciales en cascada, así como la posición ideal de los mismos (en verde). En el caso de la propuesta de V. Kazemi los regresores intermedios son *ensembles* de árboles de regresión. Esta técnica ha sido la empleada para el alineamiento facial en todos los experimentos con alineamiento no manual de este trabajo. Se ha hecho

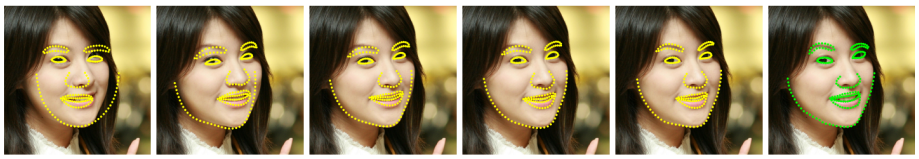


Figura 1: Estimación de puntos faciales característicos en cascada (amarillo) y posiciones ideales (verde).

uso del modelo pre-entrenado del algoritmo que proporciona la biblioteca Dlib C++ [8].

Normalización de pose Una vez se ha realizado la detección y alineamiento facial contamos con las posiciones de los puntos faciales característicos del rostro que aparece en dicha imagen. Entonces, generamos una imagen con pose normalizada girando la imagen para colocar la cara en vertical, centrando la imagen en base a los puntos característicos de nariz y ojos y recortando la imagen para encerrar de forma precisa el rostro previamente alineado.

Normalización de la iluminación Las técnicas de normalización de la iluminación buscan reducir la varianza intra-clase que presentan las tareas de reconocimiento facial en entornos no controlados. Uno de los métodos más simples para la normalización de la iluminación es la ecualización del histograma de intensidades de la imagen o *Histogram Equalization* (HE). Este método consiste en mapear el histograma de la imagen original, $H(i)$, a una distribución más uniforme. Para lograr esto se utiliza la llamada función de distribución acumulada $H'(i)$, que se calcula de la siguiente forma:

$$H'(i) = \sum_{j=0}^{i} H(j) \tag{1}$$

Una vez calculado $H'(i)$, se normaliza de tal forma que su valor máximo sea igual al valor máximo de intensidad para un *pixel* en el formato de imagen deseado. A continuación se utiliza esta función para calcular las intensidades de los *pixels* en la imagen resultante:

$$equalizada(x, y) = H'(original(x, y)) \tag{2}$$

HE ha sido utilizado para normalizar la iluminación en todos los experimentos.

3.2. Extracción de características: Local Binary Patterns

En la práctica, no resulta viable utilizar las intensidades de los *pixels* de una imagen como características para entrenar un clasificador directamente sobre

ellas. El principal motivo es que esta representación suele contener información indeseada como por ejemplo las condiciones de iluminación. Además, el número de estas características en bruto suele ser excesivamente grande. Se ha demostrado que, fijado un número de muestras de entrenamiento, el incremento de la dimensión de dichas muestras más allá de cierto límite reduce la capacidad de predicción [10].

El método elegido para la propuesta de este trabajo es el descriptor conocido como *Local Binary Patterns* (LBP) [11]. Como se verá en las secciones siguientes, su naturaleza local permite mantener las características ocluidas separadas de las extraídas de partes visibles del rostro. Además destaca por su bajo coste computacional. A continuación se proporcionan más detalles sobre el descriptor y su cómputo.

El descriptor LBP etiqueta los *pixels* de una imagen considerando la diferencia de su valor en intensidad con sus vecinos. Esta etiqueta es tratada después como un número binario. El uso de un vecindario circular y la interpolación bilineal en coordenadas de *pixel* no enteras permite utilizar el operador con cualquier radio y número de puntos en el vecindario [12]. La notación $LBP_{P,R}$ suele utilizarse para referirse al operador LBP parametrizado con R puntos o vecinos y radio de vecindario P.

En [12] se demostró que, usando el operador $LBP_{8,1}$ sobre varias imágenes, casi el 90 % de los patrones extraídos eran uniformes (su representación en binario contiene a lo sumo dos transiciones entre 0s y 1s). Por este motivo se propuso utilizar una representación en la que todos los patrones uniformes tienen su propia etiqueta, pero todos los patrones no uniformes reciben una única etiqueta. Nos referiremos a este descriptor como $LBP^{u}_{P,R}$.

Antes de entrenar un clasificador, se suele refinar la representación de LBP dividiendo la imagen en bloques de menor tamaño y calculando el número de ocurrencias de los diferentes patrones LBP en estos bloques, generando así una serie de histogramas que son concatenados para dar lugar al descriptor final. Este descriptor se conoce como *Local binary pattern histograms* (LBPH). La figura 2 muestra el proceso de extracción del descriptor LBPH en cuadrícula a partir de una imagen LBP.

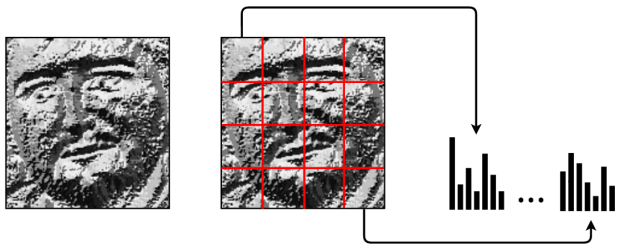


Figura 2: De izquierda a derecha: Imagen LBP, imagen LBP dividida en bloques y descriptor LBPH obtenido concatenando los histogramas de los diferentes bloques.

3.3. Identificación: un clasificador resistente a la oclusión

La propuesta fundamental de este trabajo, consiste en la definición de una modificación del algoritmo de los k-vecinos más cercanos ponderados que permita aumentar la eficacia del clasificador en situaciones de oclusión, manteniendo a la vez un coste computacional bajo.

La propuesta de este trabajo nace de la propia naturaleza del descriptor LBPH. Este descriptor extrae las características de la imagen de forma local, concatenando en ultima instancia las características que describen cada uno de los bloques del enrejado en que se divide la imagen facial. Por tanto, si somos capaces de detectar cuáles de esos bloques se encuentran ocluidos, podemos inhibir el uso de las características asociadas a dichos bloques por un clasificador.

En primer lugar, se propone un método para detectar la oclusión en bloques de la imagen analizados por LBPH, sin conocimiento *a priori* de la naturaleza de los bloques ocluidos. Definimos la *distancia local mínima* para el histograma LBP de un bloque de una imagen como la menor distancia euclídea cuadrada obtenida al comparar este histograma con los histogramas LBP de todos los bloques correspondientes a la misma posición facial en el set de entrenamiento. Entonces, la única asunción que se realiza respecto a la naturaleza de la oclusión, es que la distancia local mínima para los bloques ocluidos es normalmente mayor que para los bloques pertenecientes a imágenes sin oclusión. Para confirmar la validez de esta hipótesis podemos observar la distribución de las distancias locales mínimas para los bloques ocluidos y sin ocluir de un set de imágenes con oclusión facial (ver figura 3). Como vemos, si bien existe cierto grado de solapamiento, es posible realizar un corte conservador que deje fuera buena parte de los bloques ocluidos.

Hasta este punto se ha diseñado un método para descartar las características que no contienen información útil para realizar la clasificación. Ahora, es necesario definir un clasificador capaz de modificar dinámicamente el número de características que utilizará para emitir las predicciones, de forma que pueda aprovechar todas las características no ocluidas al tiempo que inhibe aquellas afectadas por la oclusión. Se propone una versión modificada del algoritmo de los k-vecinos más cercanos.

La versión modificada del algoritmo kNN inhibe las características ocluidas para una imagen cuya clase o identidad se quiere predecir, simplemente descartando las características inhibidas durante el cálculo de las distancias entre muestras que tiene lugar en kNN. O de forma similar: antes de calcular la distancia entre las muestras x de test y $x^{(i)}$ del set de entrenamiento, se aplica una máscara de oclusión[1] a ambas muestras de forma que las características ocluidas en x toman valor cero en x y $x^{(i)}$, y por tanto no afectan al cómputo de $d(x, x^{(i)})$. Por lo demás la versión propuesta del algoritmo kNN es similar al algoritmo original y puede combinarse con las diferentes versiones refinadas del algoritmo original como por ejemplo la de *weighted k-Nearest Neighbors* (wkNN) [13].

[1] En este contexto una máscara de oclusión consiste en un vector con ceros en las posiciones correspondientes a las características ocluidas y unos en las posiciones de las características válidas.

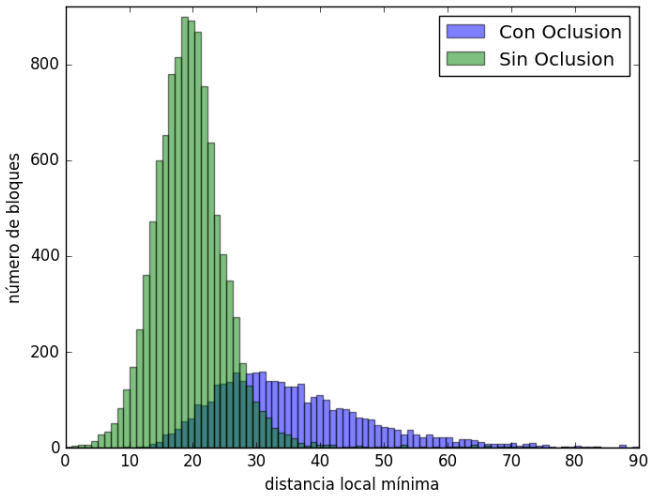


Figura 3: Distribución de la distancia mínima respecto a los bloques del set de entrenamiento para bloques ocluidos y sin oclusión.

La combinación de la técnica de detección de oclusión y clasificador con inhibición de características anteriormente descritos tiene su descripción formalizada en el algoritmo 1. Nótese que el paso 2 del algoritmo 1 fuerza el uso de la distancia euclídea cuadrada; esto se hace para reducir la complejidad del algoritmo ya que permite reutilizar las distancias locales para el cálculo de las distancias entre muestras en el paso 4. Podemos proceder de esa forma gracias a la siguiente propiedad de la distancia euclídea cuadrada: sean x e y vectores en $\mathbb{R}^d$, sea $z = x - y$ y sea $d(\cdot, \cdot)$ la distancia euclídea cuadrada, entonces:

$$\begin{aligned} d(x,y) &= ||x-y||^2 = ||z||^2 = z_1^2 + z_2^2 + \cdots + z_n^2 + z_{n+1}^2 + \cdots + z_d^2 \\ &= ||(z_1, \cdots, z_n)||^2 + ||(z_{n+1}, \cdots, z_d)||^2 \\ &= ||(x_1 - y_1, \cdots, x_n - y_n)||^2 + ||(x_{n+1} - y_{n+1}, \cdots, x_d - y_d)||^2 \\ &= d((x_1, \cdots, x_n), (y_1, \cdots, y_n)) + d((x_{n+1}, \cdots, x_d), (y_{n+1}, \cdots, y_d)) \end{aligned} \tag{3}$$

Es decir, la distancia euclídea cuadrada entre dos vectores puede obtenerse como la suma de las distancias entre los segmentos de los vectores.

Coste computacional El algoritmo wkNN se enmarca dentro de la categoría del llamado *lazy learning*, estos métodos retrasan la generalización del set de entrenamiento a la fase de evaluación o test. La complejidad computacional del algoritmo wkNN depende fundamentalmente de método usado para buscar los vecinos más cercanos, que a su vez depende de la forma en que se almacena el set de entrenamiento.

Algorithm 1 *wkNN con inhibición(S, k, p, threshold)*

1: Sea S un set de datos de entrenamiento:

$$S = \{(y^{(i)}, x^{(i)}), i = 1, 2, ..., n\}$$

con observaciones $x^{(i)} \in \mathbb{R}^d$ y etiquetas $y^{(i)}$, sea $x \in \mathbb{R}^d$ una nueva observación cuya etiqueta y se quiere predecir.

2: Calcular la matriz $n \times d/p$ de distancias locales L; para cada segmento de características $j = 1, 2, 3, \cdots, d/p$ de cada muestra $i = 1, 2, \cdots, n$ la distancia local es:

$$L_{i,j} = ||(x_{p(j-1)+1}, \cdots, x_{pj}) - (x^{(i)}_{p(j-1)+1}, \cdots, x^{(i)}_{pj})||^2$$

3: Calcular el vector que representa la máscara de oclusión estimada $M \in \mathbb{R}^{d/p}$:

$$M_j = thr(\ min(\ col_j(L)\ )\ )$$

$$thr(x) = \begin{cases} 1 & si \ \ x < threshold \\ 0 & si \ \ x > threshold \end{cases}$$

4: Encontrar los $k + 1$ vecinos más cercanos a x según la siguiente función de distancia:

$$d(x, x^{(i)}) = \sum_{j=1}^{j=d/p} M_j \cdot L_{i,j}$$

5: El vecino $(k + 1)^{th}$ se usa para estandarizar las distancias menores:

$$D_i = D(x, x^{(i)}) = \frac{d(x,x^{(i)})}{d(x,x^{(k+1)})}$$

6: Transformar las distancias normalizadas D_i con una función $K(\cdot)$ en pesos de votos $w_i = K(D_i)$.

7: Como predicción para la etiqueta de clase y de la observación x elegir la clase de la mayoría ponderada de los k vecinos más cercanos:

$$y = max_r(\sum w_i \ / \ y^{(i)} = r)$$

La aproximación más sencilla (conocida como *Naive search*), almacena las muestras de entrenamiento sin ninguna estructura adicional y realiza una búsqueda secuencial en fase de test. Por tanto, la complejidad computacional en fase de entrenamiento es $\mathcal{O}(1)$, y predecir la clase para una muestra en fase de test es $\mathcal{O}(nd+nk)$[2] donde n es la cardinalidad del set de entrenamiento, d la dimensión de las muestras y k el número de vecinos considerados [14]. Se suele considerar el hiperparámetro k como una constante, simplificando la complejidad de la fase de test a $\mathcal{O}(nd)$.

Analicemos ahora la complejidad computacional de nuestra propuesta algorítmica, wkNN con inhibición: El paso 1 del algoritmo 1 se corresponde con la fase de entrenamiento del modelo, si comparamos al propuesta con el método original vemos que la fase de entrenamiento es igual para ambos métodos, y por tanto tiene una complejidad de $\mathcal{O}(1)$. Los pasos 2 y 3 del algoritmo propuesto pueden implementarse juntos manteniendo punteros a los valores mínimos encontrados durante el cómputo de las distancias locales; esto tiene una complejidad de $\mathcal{O}(n(\frac{d}{p} \cdot p + \frac{d}{p}))$, que considerando el hiperparámetro p como una constante simplifica a $\mathcal{O}(nd)$.

El paso 4, correspondiente al cálculo de los vecinos más cercanos respecto a una métrica, tiene una complejidad computacional de $\mathcal{O}(n \cdot \frac{d}{p} + kn)$; que simplificando al considerar los hiperparámetros p y k como constantes queda en $\mathcal{O}(nd)$. Los pasos 5, 6 y 7 coinciden con los pasos finales del método original; que tienen una complejidad computacional de $\mathcal{O}(k)$ que suele despreciarse.

Entonces, la complejidad computacional total de la fase de test del algoritmo propuesto es de $\mathcal{O}(nd) + \mathcal{O}(nd)$, que simplificando queda en $\mathcal{O}(nd)$. Por tanto podemos afirmar que la complejidad computacional del algoritmo modificado es equivalente a la del algoritmo original, y de ahí que su escalabilidad es similar.

LBPH multi-escala Otra posible forma de extracción de características LBP con mayor información consiste en muestrear imágenes LBP obtenidas a diferentes escalas [15] pero en este caso haciendo uso de la tradicional cuadrícula de bloques. Se espera que la información extraída, a pesar de contener cierta redundancia, proporcione información adicional que pueda mejorar la tasa de acierto del sistema de forma significativa. La figura 4 muestra el procedimiento de extracción. El mayor problema de esta representación es su alta dimensionalidad con respecto a la versión de LBPH sobre una única escala. Si los bloques correspondientes a una misma región de la imagen (a diferentes escalas) se colocan adyacentes al concatenar los histogramas para la representación final, es posible considerarlos como una única unidad de oclusión[3] eligiendo un valor p adecuado para el algoritmo 1.

[2] Esta complejidad es para la versión del algoritmo que calcula las distancias y las almacena en un array de dimensión n. Si las distancias se re-calculan para encontrar cada vecino la complejidad es $\mathcal{O}(knd)$, que igualmente simplifica a $\mathcal{O}(nd)$.

[3] Entendemos por unidad de oclusión un conjunto de características que nuestro algoritmo clasificará como ocluidas o no ocluidas en conjunto.

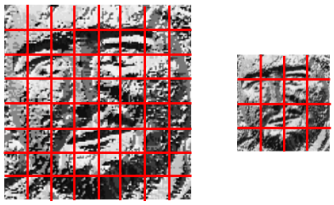


Figura 4: Extracción de características LBP multi-escala en cuadrícula.

3.4. LBPH multi-escala: uso de Random Projection

Esta sección propone una solución al problema de la alta dimensionalidad de los descriptores LBPH multi-escala. Algunos estudios proponen extraer descriptores de alta dimensionalidad, buscando maximizar la eficacia, para después aplicar alguna técnica de reducción de dimensionalidad que permita mantener la información discriminativa al tiempo que se reduce el coste computacional [16].

El método propuesto en este trabajo no es directamente compatible con las técnicas de reducción de dimensionalidad clásicas como PCA y LDA. El motivo de esta incompatibilidad radica en que necesitamos preservar la separación entre las características pertenecientes a unidades de oclusión distintas, de forma que posteriormente podamos detectar e inhibir grupos de características ocluidas.

Por tanto, la reducción de dimensionalidad ha de realizarse a nivel local. Se consideran los histogramas extraídos de la misma zona de la imagen (a diferentes escalas) como una única unidad de oclusión. Específicamente se propone aplicar el algoritmo conocido como *Random projection* (RP) ya que por sus propiedades y forma de aplicación propuesta permitirá usar descriptores LBPH multi-escala de alta dimensionalidad con una fracción del coste computacional.

RP es uno de los algoritmos de reducción de dimensionalidad lineal más simples y a la vez efectivos que se conocen. RP calcula las direcciones en base a una distribución gaussiana aleatoria. Por tanto, la matriz de proyección es independiente de los datos y su computo es extremadamente eficiente.

El principal resultado teórico que sustenta el funcionamiento de RP es el lema de Johnson-Lindenstrauss. Este resultado trata sobre la proyección con baja distorsión de puntos desde espacios euclídeos de alta dimensionalidad a espacios de baja dimensionalidad. El lema establece que un pequeño número de puntos de un espacio de alta dimensionalidad pueden ser proyectados a un espacio de mucha menor dimensionalidad de tal forma que las distancias entre las parejas de puntos sean preservadas de forma aproximada. Formalmente, dados $0 < \epsilon < 1$, un set X con n puntos en $\mathbb{R}^d$, y k un número tal que $k > 4 \cdot ln(n)/(\epsilon^2/2 - \epsilon^3/3)$ existe una función lineal $f : \mathbb{R}^d \to \mathbb{R}^k$ tal que:

$$(1-\epsilon)||u-v||^2 \leq ||f(u)-f(v)||^2 \leq (1+\epsilon)||u-v||^2 \tag{4}$$

para todo u, $v \in X$ (ver [17] para una demostración simple de este lema). Específicamente la función f puede ser calculada mediante la multiplicación por

una matriz R cuyos elementos han sido elegidos con una distribución normal. La siguiente ecuación describe el calculo de la proyección:

$$f(x) = \frac{1}{k}\, x \cdot R \tag{5}$$

A continuación se describe la forma de aplicar RP para evitar la corrupción de todas las características en caso de oclusión parcial. En primer lugar, debemos asegurarnos de que las características pertenecientes a la misma unidad de oclusión aparezcan contiguas en descriptor LBPH extraído; de esta forma el algoritmo de detección y clasificación propuesto en este trabajo será directamente aplicable. Debemos fijar el hiperparámetro p del algoritmo 1 al número de características que forman una unidad de oclusión en nuestro caso particular (Esto dependerá de los hiperparámetros elegidos para $LBP_{P,R}$ y del uso exclusivo o no de los patrones uniformes). La figura 5 muestra la forma de construir

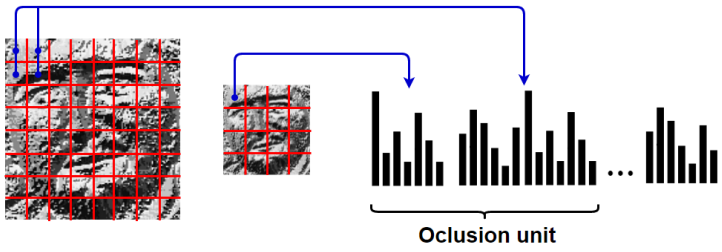


Figura 5: Las características que provienen de un área específica de la imagen se colocan juntas en el histograma resultante.

el descriptor descrita en el párrafo anterior. Una vez realizada la extracción del descriptor, podemos aplicar el algoritmo de reducción de dimensionalidad; pero debemos reducir las unidades de oclusión de forma individual. Formalmente, sea $x \in \mathbb{R}^d$ un descriptor LBPH multi-escala donde cada unidad de oclusión está formada por p características, k un número entero tal que $0 < k < p$, y R una matriz $p \times k$ generada de acuerdo a una distribución normal aleatoria. La versión reducida del descriptor x se calcula de la siguiente forma:

$$x' = f((x_1, \cdots, x_p)) \;||\; f((x_{p+1}, \cdots, x_{2p})) \;||\; \cdots \;||\; f((x_{d-p+1}, \cdots, x_d)) \tag{6}$$

donde $||$ es el operador de concatenación de vectores. Nótese que gracias al lema anteriormente descrito, para un valor k lo suficientemente grande, el resultado de ejecutar el algoritmo 1 sobre los datos reducidos es el mismo que hacerlo sobre los descriptores originales, pero con un coste computacional menor. Para demostrar esto, basta con ver qué cálculos lleva a cabo el algoritmo 1: En primer lugar se calcula la matriz de distancias locales de acuerdo con esta fórmula:

$$\begin{gathered} L_{i,j} = ||(x_{p(j-1)+1}, \cdots, x_{pj}) - (x^{(i)}_{p(j-1)+1}, \cdots, x^{(i)}_{pj})||^2 \\ \text{para} \quad i = 1, 2, \cdots, n \;\; \text{y} \;\; j = 1, 2, \cdots, d/p \end{gathered} \tag{7}$$

donde x es la nueva observación a clasificar y $x^{(i)}$ son las muestras del set de datos de entrenamiento. Si reducimos tanto la nueva observación como el set de entrenamiento de acuerdo con la ecuación 6, y aplicamos el algoritmo sobre estos datos (con el parámetro $p = k$) entonces la matriz de distancias locales calculada por el algoritmo será la siguiente:

$$L'_{i,j} = ||(x'_{k(j-1)+1}, \cdots, x'_{kj}) - (x'^{(i)}_{k(j-1)+1}, \cdots, x'^{(i)}_{kj})||^2$$
$$\text{para} \quad i = 1, 2, \cdots, n \quad \text{y} \quad j = 1, 2, \cdots, d/p \tag{8}$$

y aplicando el lema de Johnson-Lindenstrauss, tenemos que para un valor suficientemente grande de k se cumple:

$$(1 - \epsilon)\ L_{i,j} \leq L'_{i,j} \leq (1 + \epsilon)\ L_{i,j}$$
$$\text{para} \quad i = 1, 2, \cdots, n \quad \text{y} \quad j = 1, 2, \cdots, d/p \tag{9}$$

Es decir, la distorsión en los elementos de la matriz L' respecto a L queda acotada. Los siguientes pasos del algoritmo 1 se basan en esta matriz y por tanto, si la diferencia entre L' y L es suficientemente pequeña, el algoritmo proporcionará los mismos resultados cuando se ejecuta sobre los datos reducidos. En la sección de resultados experimentales se evalúa de forma empírica la eficacia de este método sobre una base de datos de imágenes parcialmente ocluidas.

4. Resultados experimentales

Esta sección tiene como objetivo presentar los resultados de los numerosos experimentos llevados a cabo durante la realización de este trabajo. Se evalúa la eficacia del algoritmo 1 sobre una base de datos de imágenes faciales con diferentes tipos de oclusión parcial y con descriptores de características de diferente tipo. La base de datos utilizada ha sido *ARFace database* [18]. Este set de datos contiene unas 4.000 imágenes a color correspondientes a 126 individuos (70 hombres y 56 mujeres). Las imágenes muestran vistas frontales de las caras de los individuos con diferentes expresiones faciales, condiciones de iluminación y oclusiones parciales (con gafas de sol y bufandas). Se proporciona la versión manualmente alineada de todas las fotos.

De todas estas imágenes se ha seleccionado un subconjunto para formar varios sets de imágenes para nuestros experimentos. Se ha formado un set de entrenamiento, uno de validación y varios de test con las siguientes características:

- *Training set*: Set formado por una imagen por individuo (neutral, iluminación uniforme, primera sesión).
- *Validation set*: Set formado por una imagen de la mayoría de individuos[4] (neutral, iluminación uniforme, segunda sesión).
- *Illumination test set*: Set formado por casi cuatro imágenes por individuo (neutral, iluminación izquierda/derecha, primera y segunda sesión).

[4] Las imágenes de la segunda sesión no están disponibles para todos los individuos.

- *Glasses test set*: Set formado por casi dos imágenes por individuo (gafas de sol, iluminación uniforme, primera y segunda sesión).
- *Scarf test set*: Set formado por casi dos imágenes por individuo (bufanda, iluminación uniforme, primera y segunda sesión).

Se compara la tasa de acierto del algoritmo propuesto con la obtenida usando otros clasificadores comunes en el ámbito del reconocimiento facial, específicamente *Logistic Regression* (LR), *Support vector machine* (SVM) y *Naive Bayes* (consultar [19] para más información sobre estos métodos). Solo están disponibles para el entrenamiento de los modelos una o como mucho dos imágenes para cada individuo. Por tanto, no tiene sentido parametrizar los algoritmos de clasificación de vecinos más cercanos con valores de k mayores que uno, ya que en cualquier caso la predicción emitida dependerá solo del vecino más cercano.

Sin embargo, la versión modificada del algoritmo wkNN tiene más hiperparámetros que deben ser ajustados adecuadamente: El parámetro p del algoritmo 1 determina el tamaño en número de características consecutivas de cada unidad de oclusión. Este parámetro viene determinado directamente por el tipo de descriptor LBP utilizado y en cada caso de estudio se proporciona su valor.

El parámetro restante es el *threshold* o valor de corte, en un escenario ideal dispondríamos de un set de imágenes con oclusión parcial para ajustar este valor, pero uno de los objetivos de este trabajo ha sido que el método propuesto no necesitase de imágenes con oclusión para su entrenamiento. Afortunadamente es posible encontrar un buen valor para este parámetro mediante un set de datos de validación con imágenes sin oclusión, incluso con menos de una imagen adicional por cada individuo. El procedimiento a seguir será el siguiente:

1. Se elige un valor suficientemente elevado para el *threshold* (para valores de threshold suficientemente elevados el método propuesto se comporta como wkNN, podemos usar esto para determinar si comenzamos la selección del hiperparámetro desde un valor suficientemente elevado).
2. Se entrena el algoritmo 1 sobre el *Training set* y se evalúa sobre el *Validation set*.
3. Decrementamos el *threshold* y repetimos el paso 2 hasta que se produzca un cambio significativo en la tasa de acierto del modelo. Esto nos indicará que algunos de los bloques no ocluidos, con información útil, han sido erróneamente clasificados como ocluidos, así que debemos fijar el *threshold* en el valor inmediatamente superior al actual.

El procedimiento para evaluar el método propuesto para cada tipo de descriptor ha sido por tanto el siguiente: Se estima el *threshold* mediante el Trainig set y *Validation set* (conforme a lo descrito anteriormente), se entrenan los modelos ya parametrizados sobre las imágenes del *Training set* y el *Validation set* y finalmente se evalúan los modelos sobre los diferentes sets de testeo disponibles.

Resultados para LBPH en cuadrícula y alineamiento automático Se usan características del tipo $LBP^{u}_{8,2}$ extraídas mediante una cuadrícula uniforme. La dimensión de la cuadrícula ha sido establecida a un valor típico de 8×8.

La figura 6 muestra los resultados de validación para wkNN y para el método propuesto con diferentes valores de *threshold*. En virtud de los resultados de validación se fijó el valor de *threshold* para wkNN con inhibición en 27. Se entrenaron entonces los modelos a evaluar sobre el *Training set* y el *Validation set* y se evaluaron sobre los diferentes sets de testeo; los resultados se muestran en la tabla 1.

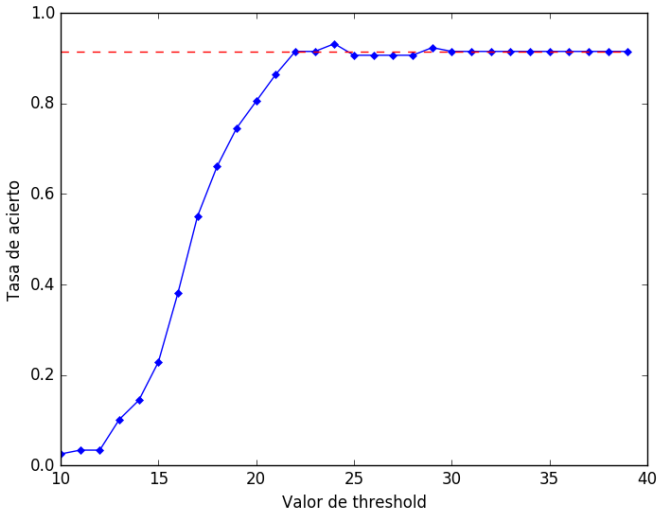


Figura 6: Resultados de validación del método wkNN con inhibición para diferentes valores de threshold (azul) y resultado de validación para wkNN clásico (rojo).

Tabla 1: Tasas de acierto para LBPH en cuadrícula con alineamiento facial automático.

Método	**Illumination**	**Scarf**	**Glasses**
wkNN estándar	81.4 %	39.4 %	30.0 %
wkNN con inhibición $p = 59$; *threshold* $= 27$	**96.2** %	**83.6** %	**50.2** %
SVM (polynomial kernel)	78.1 %	36.9 %	25.1 %
Logistic Regression	84.8 %	45.0 %	23.4 %
Naive Bayes (Multinomial)	82.5 %	43.7 %	20.1 %

Resultados para LBPH en cuadrícula multi-escala y alineamiento automático Se usan características del tipo $LBP^u_{8,2}$ extraídas mediante cuadrículas uniformes a varias escalas. En primer lugar se aplicó una cuadrícula de 12×12, posteriormente se redujeron las imágenes a la mitad de tamaño y se aplicó una cuadrícula de 6×6. La figura 7 muestra los resultados de validación para wkNN con inhibición y para wkNN con diferentes valores de *threshold*. En base a estos resultados se fijó el valor de *threshold* en 17. Se entrenaron entonces los modelos a evaluar sobre la suma del *Training set* y el *Validation set* y se evaluaron sobre los diferentes sets de testeo; los resultados se muestran en la tabla 2.

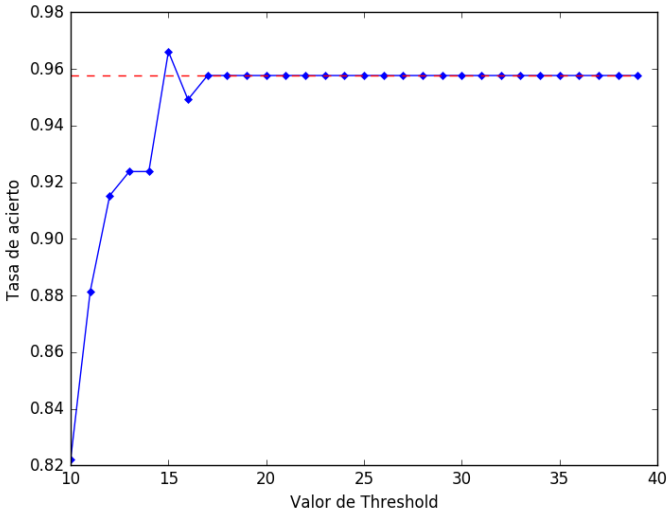


Figura 7: Resultados de validación del método wkNN con inhibición para diferentes valores de threshold (azul) y resultado de validación para wkNN clásico (rojo).

Si se comparan los resultados correspondientes al uso del descriptor LBPH multi-escala y simple, se observa un incremento significativo de la tasa de acierto. Sin embargo, como ya se ha explicado, la principal desventaja de este descriptor a varios niveles es su dimensionalidad excesiva. En este caso cada histograma tiene 59 características (por usar $LBP^u_{8,2}$), que multiplicadas por los $12 \cdot 12 + 6 \cdot 6$ histogramas que forman en este caso el descriptor de una imagen, dan un total de 10.620 características.

Resultados para LBPH en cuadrícula multi-escala con Random Projection y alineamiento automático Se usan características del tipo $LBP^u_{8,2}$

Tabla 2: Tasas de acierto para LBPH en cuadrícula multi-escala con alineamiento facial automático.

Método	**Illumination test**	**Scarf test**	**Glasses test**
wkNN estándar	98.8 %	73.3 %	34.9 %
wkNN con inhibición $p = 59$; *threshold* $= 17$	**99.2** %	**89.2** %	**50.6** %
SVM (polynomial kernel)	88.1 %	59.6 %	27.9 %
Logistic Regression	96.2 %	75.1 %	28.3 %
Naive Bayes (Multinomial)	86.29 %	72.1 %	37.03 %

extraídas mediante cuadrículas uniformes a varias escalas. En primer lugar se aplicó una cuadrícula de 12×12, posteriormente se redujeron las imágenes a la mitad de tamaño y se aplicó una cuadrícula de 6×6. Finalmente se aplicó la técnica de reducción de dimensionalidad a nivel local tal como se explica en el apartado 3.4.

En este caso, se redujo casi a la mitad la dimensionalidad de cada unidad de oclusión. Las unidades de oclusión antes de reducir constaban de cinco histogramas con 59 características cada uno (295 por unidad). Fueron reducidas a 150 características por unidad de oclusión. Pasamos así de trabajar con descriptores de imagen de 10.620 características a 5.400 características. Recordemos que al reducir la dimensionalidad es necesario considerar los histogramas extraídos de una misma región de la imagen como una única unidad de oclusión, para más detalles consultar la sección 3.4. La figura 8 muestra los resultados de validación para wkNN con inhibición y para wkNN con diferentes valores de *threshold*. Se fijó el valor de *threshold* en 100. Se entrenaron entonces los modelos a evaluar sobre la suma del *Training set* y el *Validation set* y se evaluaron sobre los diferentes sets de testeo; los resultados se muestran en la tabla 3.

Tabla 3: Tasas de acierto para LBPH en cuadrícula multi-escala y RP con alineamiento facial automático.

Método	**Illumination test**	**Scarf test**	**Glasses test**
wkNN estándar	98.5 %	66.5 %	31.2 %
wkNN con inhibición $p = 150$; *threshold* $= 100$	**98.8** %	**90.5** %	**51.0** %
SVM (polynomial kernel)	85.5 %	55.3 %	20.9 %
Logistic Regression	93.3 %	69.0 %	25.5 %
Naive Bayes (Bernoulli)	84.0 %	54.5 %	27.1 %

Si se comparan los resultados de este apartado con el anterior, vemos que a pesar de que hemos reducido a la mitad la dimensión del descriptor utilizado, las tasas de acierto se han mantenido casi intactas para el caso de wkNN con inhibición.

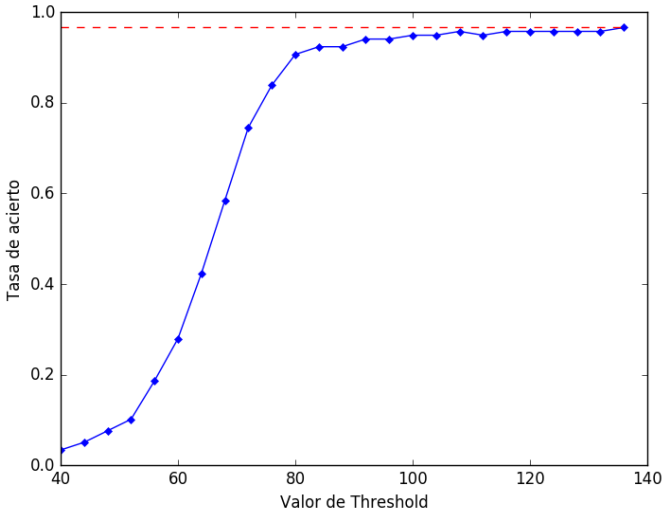


Figura 8: Resultados de validación del método wkNN con inhibición para diferentes valores de threshold (azul) y resultado de validación para wkNN clásico (rojo).

Resultados para LBPH en cuadrícula y alineamiento manual Se usan características del tipo $LBP^{u}_{8,2}$ extraídas mediante una cuadrícula uniforme sobre las imágenes. La dimensión de la cuadrícula ha sido establecida a un valor típico de 8×8. La figura 9 muestra los resultados de validación para wkNN con inhibición y wkNN para diferentes valores de *threshold*. El valor de *threshold* se fijó en 30. Se entrenaron entonces los modelos a evaluar sobre la suma del *Training set* y el *Validation set* y se evaluaron sobre los diferentes sets de testeo; los resultados se muestran en la tabla 4.

Tabla 4: Tasas de acierto para LBPH en cuadrícula con alineamiento facial manual.

Método	**Illumination test**	**Scarf test**	**Glasses test**
wkNN estándar	95.5 %	76.5 %	69.5 %
wkNN con inhibición $p = 59$; *threshold* $= 30$	**99.5** %	**91.5** %	**83.5** %
SVM (polynomial kernel)	96.5 %	75.0 %	61.0 %
Logistic Regression	98.5 %	81.0 %	68.0 %
Naive Bayes (Multinomial)	94 %	76.5 %	69.0 %

Como se aprecia en la tabla, wkNN con inhibición supera a los métodos estándar en todos los sets de datos de evaluación, además las puntuaciones en

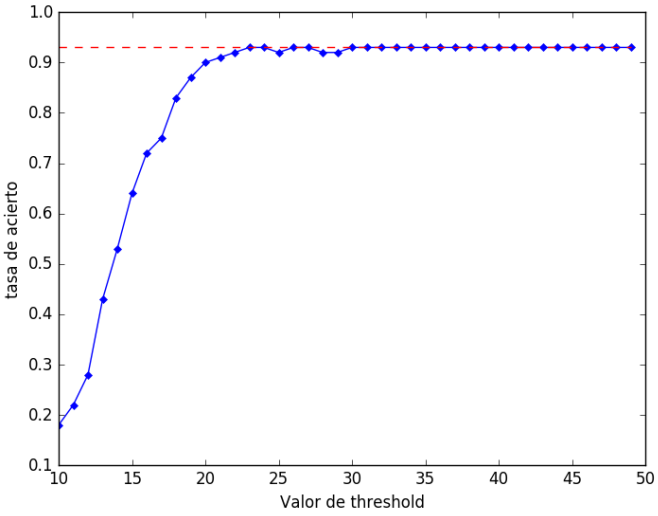


Figura 9: Resultados de validación del método wkNN con inhibición para diferentes valores de threshold (azul) y resultado de validación para wkNN clásico (rojo).

los casos de oclusión son significativamente más altas que en los experimentos anteriores (los de alineación automática). Esto demuestra el enorme impacto de la oclusión sobre los métodos de alineamiento automático.

Resultados para LBPH en cuadrícula multi-escala y alineamiento manual Se usan características del tipo $LBP^{u}_{8,2}$ extraídas mediante cuadrículas uniformes sobre las imágenes a varias escalas. En primer lugar se aplicó una cuadrícula de 12×12, posteriormente se redujeron las imágenes a la mitad de tamaño y se aplicó una cuadrícula de 6×6. La figura 10 muestra los resultados de validación para wkNN con inhibición y para wkNN con diferentes valores de *threshold*. Se fijó el valor de *threshold* en 16. Se entrenaron entonces los modelos a evaluar sobre la suma del *Training set* y el *Validation set* y se evaluaron sobre los diferentes sets de testeo; los resultados se muestran en la tabla 5.

Resultados para LBPH en cuadrícula multi-escala con Random Projection y alineamiento manual Se usan características del tipo $LBP^{u}_{8,2}$ extraídas mediante cuadrículas uniformes sobre las imágenes a varias escalas. En primer lugar se aplicó una cuadrícula de 12×12, posteriormente se redujeron las imágenes a la mitad de tamaño y se aplicó una cuadrícula de 6×6. Finalmente se aplicó una técnica de reducción de dimensionalidad a nivel local tal como se explica en el apartado 3.4.

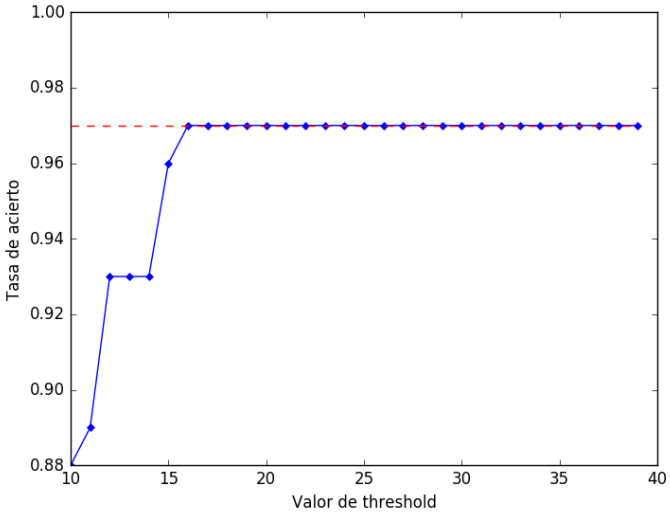


Figura 10: Resultados de validación del método wkNN con inhibición para diferentes valores de threshold (azul) y resultado de validación para wkNN clásico (rojo).

Tabla 5: Tasas de acierto para LBPH en cuadrícula multi-escala con alineamiento facial manual.

Método	**Illumination test**	**Scarf test**	**Glasses test**
wkNN estándar	**100.0** %	92.5 %	90.0 %
wkNN con inhibición $p = 59$; *threshold* $= 16$	99.5 %	**96.5** %	**93.5** %
SVM (polynomial kernel)	**100.0** %	92.0 %	84.5 %
Logistic Regression	**100.0** %	93.0 %	89.5 %
Naive Bayes (Multinomial)	98.5 %	93.5 %	89.5 %

En este caso, se redujo casi a la mitad la dimensionalidad de cada unidad de oclusión. Las unidades de oclusión antes de reducir constaban de cinco histogramas con 59 características cada uno (295 por unidad). Fueron reducidas a 150 características por unidad de oclusión. Pasamos así de trabajar con descriptores de imagen de 10.620 características a 5.400 características. Para más detalles consultar la sección 3.4. La figura 11 muestra los resultados de validación para wkNN con inhibición y wkNN para diferentes valores de *threshold*. Se fijó el valor de *threshold* en 111. Se entrenaron entonces los modelos a evaluar sobre la suma del *Training set* y el *Validation set* y se evaluaron sobre los diferentes sets de testeo; los resultados se muestran en la tabla 6.

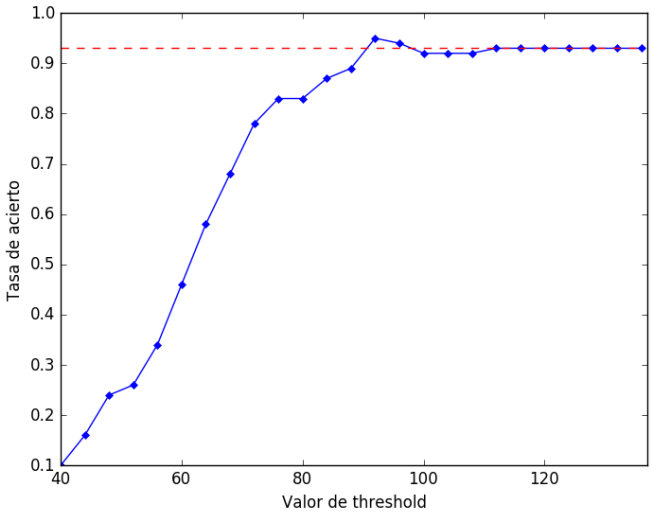


Figura 11: Resultados de validación del método wkNN con inhibición para diferentes valores de threshold (azul) y resultado de validación para wkNN clásico (rojo).

5. Conclusiones

Las principales conclusiones que se derivan de los resultados teóricos y experimentales de este trabajo son las siguientes:

- Se ha evaluado la mejora que el método propuesto permite respecto a diversos clasificadores clásicos sobre un conocido set de imágenes con oclusión parcial. Los resultados demuestran que el método propuesto supera al tradicional en casi todos los casos.

Tabla 6: Tasas de acierto para LBPH en cuadrícula multi-escala y RP con alineamiento facial manual.

Método	**Illumination test**	**Scarf test**	**Glasses test**
wkNN estándar	**100** %	92.0 %	86.0 %
wkNN con inhibición $p = 150$; *threshold* $= 111$	99.5 %	**97.0** %	**92.0** %
SVM (polynomial kernel)	**100.0** %	92.0 %	84.5 %
Logistic Regression	**100.0** %	93.0 %	89.5 %
Naive Bayes (Bernoulli)	95.5 %	93.5 %	89.0 %

- Se ha analizado la complejidad computacional del método propuesto, llegando a la conclusión de que su complejidad computacional es equivalente a la de wkNN, es decir: $\mathcal{O}(d \cdot n)$.
- Se ha propuesto, justificado de forma teórica y evaluado un método para reducir la dimensionalidad de los descriptores LBPH extraídos a varias escalas que es compatible con el método de clasificación propuesto.
- Se ha demostrado que los métodos de alineamiento automático son sensibles a las condiciones de oclusión, y que esto puede tener un severo impacto negativo sobre la tasa de acierto del sistema. El método propuesto en este trabajo demuestra ser menos sensible a estos errores de alineamiento que los clasificadores clásicos.

En base a los resultados experimentales y las conclusiones que de ellos se han extraido, las lineas de trabajo futuro que se proponen son las siguientes:

- Diseño de otros algoritmos con inhibición de características ocluidas: Otros algoritmos de clasificación pueden ser adaptados para permitir la inhibición de las características que se detecten como ocluidas. En especial, puede ser interesante trabajar con redes neuronales artificiales del tipo *Locally connected*. Sería interesante estudiar también el uso del *dropout* para facilitar la emisión de predicciones correctas cuando algunas de las características no están disponibles.
- Estudio de la aplicabilidad de las técnicas desarrolladas a otros descriptores de características locales, llevando a cabo un estudio comparativo.
- Diseño de técnicas de alineamiento facial resistentes a la oclusión: Nuestros experimentos han confirmado el fenómeno ya descrito en la literatura: gran parte de la pérdida de precisión en sistemas de reconocimiento facial frente a la oclusión se debe a los errores de alineamiento.

Referencias

1. H. K. Ekenel, *A robust face recognition algorithm for real-world applications*. PhD thesis, Karlsruhe, Univ., Diss., 2009, 2009.
2. A. M. Martínez, "Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 6, pp. 748–763, 2002.

3. X. Tan, S. Chen, Z.-H. Zhou, and F. Zhang, "Recognizing partially occluded, expression variant faces from single training image per person with som and soft k-nn ensemble," *Neural Networks, IEEE Transactions on*, vol. 16, no. 4, pp. 875–886, 2005.
4. R. Min, A. Hadid, and J.-L. Dugelay, "Improving the recognition of faces occluded by facial accessories," in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pp. 442–447, IEEE, 2011.
5. H. Jia and A. M. Martinez, "Face recognition with occlusions in the training and testing sets," in *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, pp. 1–6, IEEE, 2008.
6. J. Liu, Y. Deng, and C. Huang, "Targeting ultimate accuracy: Face recognition via deep embedding," *arXiv preprint arXiv:1506.07310*, 2015.
7. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, IEEE, 2005.
8. D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.
9. G. Tzimiropoulos, "Project-out cascaded regression with an application to face alignment," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 3659–3667, IEEE, 2015.
10. G. P. Hughes, "On the mean accuracy of statistical pattern recognizers," *Information Theory, IEEE Transactions on*, vol. 14, no. 1, pp. 55–63, 1968.
11. T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.
12. T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.
13. K. Hechenbichler and K. Schliep, "Weighted k-nearest-neighbor techniques and ordinal classification," tech. rep., Discussion paper//Sonderforschungsbereich 386 der Ludwig-Maximilians-Universität München, 2004.
14. R. Weber, H.-J. Schek, and S. Blott, "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces," in *VLDB*, vol. 98, pp. 194–205, 1998.
15. C.-H. Chan, J. Kittler, and K. Messer, *Multi-scale local binary pattern histograms for face recognition*. Springer, 2007.
16. D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3025–3032, 2013.
17. S. Dasgupta and A. Gupta, "An elementary proof of a theorem of johnson and lindenstrauss," *Random Structures & Algorithms*, vol. 22, no. 1, pp. 60–65, 2003.
18. A. M. Martinez, "The ar face database," *CVC Technical Report*, vol. 24, 1998.
19. C. M. Bishop, "Pattern recognition," *Machine Learning*, 2006.

Algoritmos y Herramientas para Composición Automática de Melodías

Diego Milla de Castro, Belén Pérez Lancho y María Navarro Cáceres

Departamento de Informática y Automática, Universidad de Salamanca

Resumen Esta investigación explora las posibilidades de la composición inteligente de melodías controlada mediante dispositivos de tipo *joystick*. Se estudia el uso de los dos ejes disponibles en estos dispositivos para dirigir el tono y la duración de las notas de la melodía generada. La investigación realizada parte de un conjunto de ficheros MIDI de los que se extraen las notas con sus respectivas duraciones para entrenar un modelo de Markov. Se propone un algoritmo de control para modificar las transiciones presentes en este modelo en función de la posición del dispositivo, y se compone una melodía en tiempo real en función estas probabilidades modificadas. También se exponen los resultados de diversos experimentos realizados para estudiar el impacto de las diferentes posibilidades ofrecidas por el entrenamiento del modelo y el algoritmo de control sobre la generación de la melodía.

1. Introducción y Estado del Arte

Los diferentes avances computacionales y en el campo de la Inteligencia Artificial que han ocurrido a lo largo de los últimos a'nos han llamado la atención de investigadores con todo tipo de orígenes y motivaciones, creando de este modo campos innovadores que unen conceptos aparentemente tan dispares como, por ejemplo, la Inteligencia Artificial y el Arte. De estas dos disciplinas nace el área de la Creatividad Computacional (también conocida como Creatividad Artificial), que puede ser vagamente definida como el análisis computacional y/o la síntesis de obras de arte, de un modo parcial o completamente automatizado [4]. Dada la particular naturaleza de este campo de investigación, que junta dos sectores con métodos y objetivos muy distintos (a veces incluso opuestos), el estado del arte es muy diverso y difícil de comparar.

Sin embargo, recientemente se está desarrollando de forma significativa el área de la Creatividad Computacional con la entrada de nuevos actores tan importantes como *Google*, con proyectos como *Deep Dream* ([5], una red neuronal inversa que transforma imágenes) o más recientemente este mes de Junio con el anuncio de *Magenta* ([6], el equivalente para la generación de música). Sin llegar al extremo de una generación completamente automatizada, también emerge la posibilidad de utilizar algoritmos para controlar la generación de arte mediante dispositivos de control que permitan indicar características de un nivel lo suficientemente abstracto para dirigir el proceso de creación. Es el caso, por ejemplo, de *MotionComposer* ([3]): una investigación de 2015 en la que utilizan

un dispositivo para componer música a partir del movimiento de personas con discapacidad.

Este proyecto explora las posibilidades de la generación inteligente de melodías controlada por este tipo de dispositivos. Se propone la utilización del dispositivo para controlar las variaciones de tono y duración de las notas de la melodía generada. Esta investigación requiere un conjunto de tareas muy variadas, por lo que sus objetivos son bastante diversos: seleccionar y procesar los datos de entrada, obtener un modelo que los represente, desarrollar un algoritmo de generación de melodías a partir de este modelo que pueda ser "dirigido" por el usuario, implementar el proceso de comunicación con el dispositivo, ofrecer una realimentación al usuario tanto visual como a través del propio dispositivo, y realizar diversos experimentos para estudiar los resultados obtenidos.

La metodología propuesta parte de unos ficheros MIDI ([8]) de referencia de los que se extraen las melodías para entrenar una cadena de Markov. Este modelo es el encargado de la parte "inteligente" de la generación melodías ya que es el que determina cuáles son las diferentes notas posibles para cada punto de la melodía. Dentro de estas posibilidades, la interacción mediante el dispositivo permite dirigir la generación de la melodía modificando las probabilidades de las notas presentes en la cadena de Markov. Se ha desarrollado una aplicación que permite experimentar con esta metodología, generando la melodía y dibujando dos gráficos en tiempo real para poder visualizar los resultados. A lo largo de este artículo se detallan las diferentes etapas de la investigación realizada y se expone una serie de experimentos que permiten estudiar el efecto sobre la generación de la melodía de las diferentes posibilidades ofrecidas por el entrenamiento del modelo y el algoritmo de control.

2. Metodología

Para poder realizar esta investigación han sido necesarias diversas etapas que van desde la obtención y extracción de los datos hasta la visualización y el análisis de los resultados y que se explican en detalle en los diferentes apartados de esta sección. El esquema de la figura 1 permite hacerse una idea general de las diferentes etapas que se llevan a cabo para la generación de la melodía dirigida mediante el dispositivo. El primer paso es la obtención y el procesamiento de los datos de entrada que se detalla en el apartado 2.1 y, a continuación, se entrena una cadena de Markov de la forma especificada en la parte 2.2. En la subsección 2.3 se incluye toda la información relacionada con el dispositivo. El proceso de generación de la melodía a partir de la cadena de Markov y de la posición del dispositivo se trata en el apartado 2.4. Por último, en la subsección 2.5 se explica el proceso de realimentación y visualización de resultados.

Al tratarse de una investigación bastante particular en un campo muy heterogéneo, no existe ninguna herramienta disponible que ofrezca la libertad necesaria para realizar este proyecto por lo que ha sido necesario desarrollarla. De todas las opciones disponibles, se ha elegido el lenguaje de programación *Scala* ([10]). El principal argumento detrás de esta decisión es la posibilidad de utilizar los completos paquetes MIDI de *Java* (*"javax.sound.midi"*)

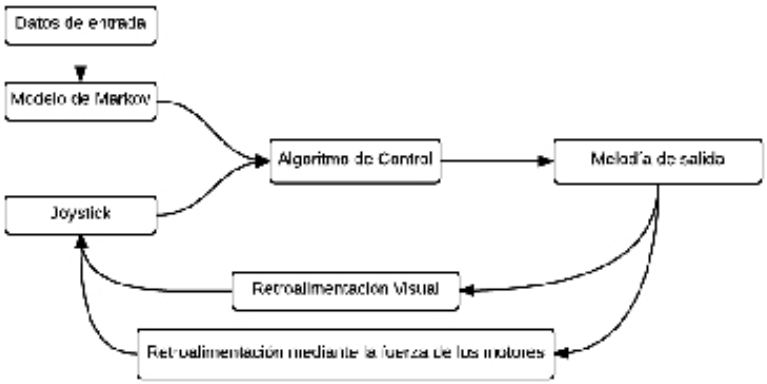


Figura 1: Generación inteligente controlable de melodías

desde un punto de vista más abstracto que ofrece más posibilidades y permite un desarrollo más rápido que programando directamente en *Java*. Aprovechando la flexibilidad disponible al implementar la herramienta desde 0, se ha podido integrar todo el proceso de investigación en una misma aplicación. Se puede consultar el código fuente de la aplicación en *GitHub*, en la dirección `https://github.com/dmilla/kMMG-Final`.

2.1. Obtención y Procesamiento de Datos

Siendo el objetivo generar y controlar una melodía variando tanto el tono como el ritmo, se necesitan datos de entrada que representen estas dos características. Actualmente, no existe un formato que trate únicamente estas dos variables por lo que es necesario extraer los datos de otro tipo de archivos. Se ha decidido utilizar ficheros MIDI (*Musical Instrument Digital Interface*, [8]) ya que contienen la información necesaria de forma bien estructurada y fácilmente accesible. Además, son archivos bastante ligeros ya que permiten codificar una canción completa en unos cientos de líneas, por ejemplo en algunos *kilobytes*. Esto se debe a que no registran el sonido en sí, ya que contienen las instrucciones (como si fuera una partitura) que permiten reconstruir la canción utilizando un secuenciador y un sintetizador que trabajen con las especificaciones MIDI. La figura 2 resume el proceso de obtención y procesamiento de datos que se detalla a lo largo de esta sección

Una vez seleccionados los ficheros MIDI con los que se va a experimentar, el siguiente paso es extraer la información necesaria para el proyecto: las notas y su duración. Según las especificaciones MIDI ([8]), cada fichero MIDI representa una secuencia (que se corresponde generalmente con una canción o composición) que a su vez se compone de una o varias pistas. Estas pistas se caracterizan

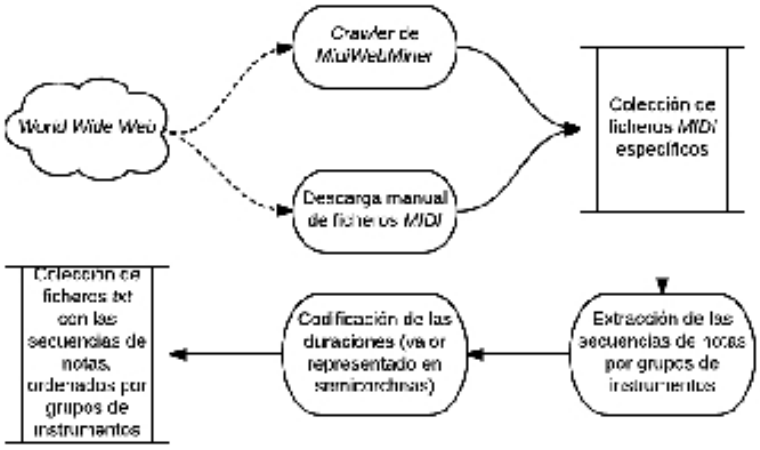


Figura 2: Obtención y procesamiento de datos

por una sucesión de eventos MIDI (mensajes MIDI asociados a un tiempo en particular).

Dadas estas especificaciones del formato MIDI, se pueden extraer fácilmente los datos necesarios mediante unos pocos bucles. Primero, se itera entre todos los ficheros MIDI objetivo para obtener la secuencia correspondiente. A continuación, se itera entre las diferentes pistas de la secuencia, y una vez dentro de cada pista se itera entre los diferentes eventos de cada una. Para este proyecto se necesitan las notas y su duración, por lo que nos interesan únicamente los eventos de tipo *NOTE_ON* y *NOTE_OFF* ([1]). Estos eventos sirven para determinar el inicio y el fin de una determinada nota, especificada en el primer byte del mensaje MIDI. El segundo byte del mensaje indica la intensidad de la nota (denominada "velocidad" en los mensajes MIDI), y para este proyecto se ignora excepto en el caso de que sea un mensaje de tipo *NOTE_ON* con velocidad 0, que es el equivalente a un mensaje de tipo *NOTE_OFF* ([7]). También se tienen en cuenta los eventos de tipo *PROGRAM_CHANGE*, que indican un cambio de instrumento, para extraer las notas por grupos de instrumentos ya que las melodías están destinadas a un instrumento en particular y de este modo se pueden seleccionar únicamente aquellas que correspondan con el grupo de instrumentos con el que se desee experimentar.

Para determinar la duración de una determinada nota, cada vez que se encuentra un evento *NOTE_ON* se itera entre los eventos siguientes hasta encontrar el evento *NOTE_OFF* (o *NOTE_ON* con velocidad 0) que determina su fin. Cada secuencia MIDI se caracteriza por una resolución que determina el número de ticks por nota negra. Mediante la diferencia de la posición de ambos eventos en la pista, podemos determinar su duración en "ticks" MIDI. Para este proyecto la duración mínima de las notas es de una semicorchea (un cuarto de

negra), por lo que se normaliza la duración de cada nota a su duración más cercana en semicorcheas y se guarda como el número de estas. Las notas se guardan por su valor MIDI (un byte, es decir un número de 0 a 127 que se corresponde con una nota de una octava en particular). Los silencios (tiempos entre el final de una nota y el inicio de la siguiente) se consideran como una nota más, en este caso se les ha asociado el valor -1, y también se les asocia una duración en semicorcheas.

Tras realizar todo este proceso, se exportan los resultados a una colección de ficheros de texto en los que se encuentran las secuencias de notas con sus respectivas duraciones en semicorcheas. Por ejemplo, una nota negra Do en la segunda octava se representaría "(48,4)": es decir, la nota 48 durante 4 semicorcheas. Si le a'nadimos una corchea Fa en la misma octava a continuación, la secuencia se representaría "(48,4) - (53,2)". De este modo se guardan los datos en ficheros fácilmente accesibles si se desean utilizar de nuevo sin necesidad de volver a realizar el proceso de extracción.

2.2. Elección y Entrenamiento del Modelo

Las cadenas de Markov son una herramienta ampliamente utilizada para modelizar las propiedades temporales de diversos fenómenos, desde la estructura de un texto hasta fluctuaciones económicas. Al tratarse de modelos relativamente fáciles de generar, también se utilizan para aplicaciones de generación de contenido, como la generación de textos o de música (por ejemplo en [9]). El enfoque tradicional de los algoritmos de generación de secuencias de Markov suele ser muy rígido y de tipo "de izquierda a derecha"([11]), propiedades que los hacen fundamentalmente inadaptados a un control interactivo como el que se plantea en este proyecto. Sin embargo, las propiedades de las cadenas de Markov siguen siendo muy pertinentes para la generación inteligente de melodías, por lo que se ha decidido usar este tipo de modelo pero utilizando un algoritmo de generación de secuencias diferente y adaptado al control interactivo.

Dado que se dispone de un espacio de control limitado, el modelo se entrena con datos normalizados de forma que pertenezcan al espacio de control disponible. Por un lado, las notas se normalizan a su equivalente en las tres primeras octavas (a un valor entre 0 y 35). Se ha desarrollado un algoritmo que determina cuáles son las tres octavas consecutivas que reagrupan el mayor número de notas de los datos de entrada y que serán consideradas las octavas de referencia para los datos correspondientes. Las notas por debajo de esta referencia se trasladan a la octava de referencia inferior, y aquellas por encima a la octava de referencia superior. De este modo todas las notas se sitúan dentro de las octavas de referencia, que se normalizan a su vez a su valor correspondiente en las tres primeras octavas para mayor simplicidad y generalización. Sin embargo, la melodía se generará a continuación respetando las octavas de referencia, ya que también se ha implementado un parámetro de normalización de la melodía de salida que se ajusta automáticamente en función de la normalización de los datos de entrada.

Por otro lado, las duraciones se normalizan a 8 duraciones posibles: semicorchea, corchea, corchea con puntillo, negra, negra con puntillo, blanca, blanca con

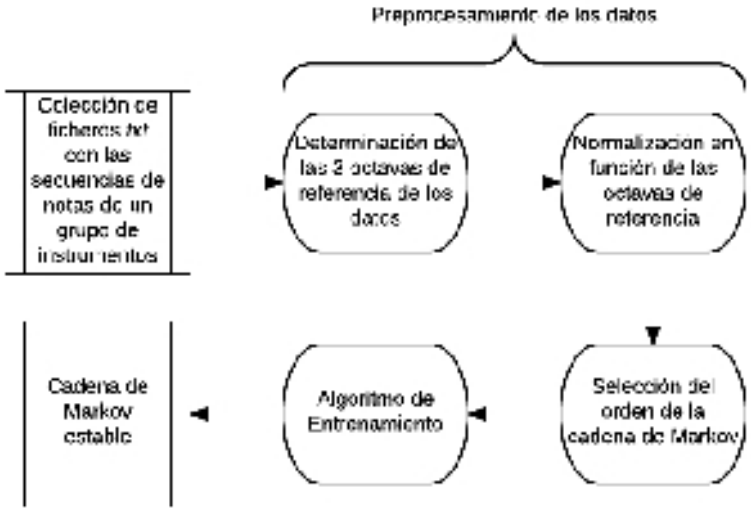


Figura 3: Entrenamiento de la cadena de Markov

puntillo y redonda (un puntillo equivale a multiplicar por 1,5 la duración de la nota). De este modo las 8 duraciones posibles expresadas en semicorcheas son 1, 2, 3, 4, 6, 8, 12 y 16.

La cadena de Markov se construye a continuación mediante un sencillo proceso de entrenamiento: se recorren todas las secuencias de notas con duración extraídas para un determinado grupo de instrumentos, actualizando constantemente el estado de la cadena de Markov según va iterando el algoritmo y a'nadiendo progresivamente cada transición a su estado correspondiente. Una vez finalizado el bucle, para asegurar la estabilidad de la cadena de Markov (es decir, que exista al menos una transición para cada estado), se sigue iterando desde el principio de los datos (pero esta vez con las notas del final de la secuencia como estado) un número de veces igual al orden del modelo que se esté entrenando. El esquema de la figura 3 muestra las diferentes etapas expuestas previamente.

2.3. Dispositivo

Se ha utilizado un dispositivo de tipo *joystick*' disponible en el laboratorio del centro de I+D de la *USAL*. El principal investigador detrás de este dispositivo es el profesor Wataru Hashimoto del OIT (*Osaka Institute of Technology*). Técnicamente, el dispositivo está compuesto por 2 motores anclados a un marco de aluminio e interconectados formando un pantógrafo con 2 grados de libertad

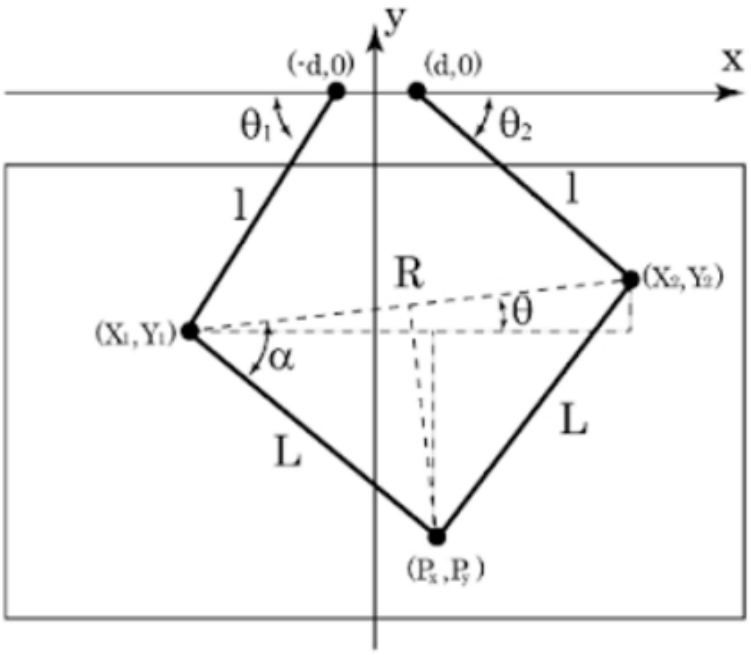


Figura 4: Plano de coordenadas del dispositivo

de tal forma que cada par de ángulos de entrada de los motores ($\theta 1, \theta 2$) se corresponde con un punto en particular en el plano esquematizado en la figura 4. Una de las ventajas de esta configuración es que los motores pueden ejercer una fuerza de retroalimentación que puede ser de utilidad para el usuario en diversas aplicaciones, como en el caso de este proyecto por ejemplo.

Dados la distancia desde el centro hasta los motores (d) y la longitud de los "brazos" superior e inferior (l y L respectivamente), el sistema forma un conjunto de triángulos determinados por los ángulos ($\theta 1, \theta 2$). Esto permite la obtención de las coordenadas (Px, Py) mediante un sistema de ecuaciones trigonométricas. Para aplicar una fuerza de retroalimentación se ha empleado el algoritmo de un proyecto con el mismo dispositivo realizado anteriormente y que permite calcular los valores que se envían a los motores para simular un determinado vector de fuerza en función de la torsión ejercida.

Este dispositivo trabaja con un espacio de coordenadas bastante peculiar. Por ejemplo, en el plano (x, y) esquematizado en la figura 4 se llegan a alcanzar valores superiores a 45 en el eje X e inferiores a 55 en el eje Y. Sin embargo, no se pueden alcanzar las "esquinas" equivalente de coordenadas (45, -55) ya que la longitud de los brazos lo impide. Además, en la práctica, algunos puntos del espacio original también son dificilmente accesibles debido a la construcción del dispositivo (choques con el aluminio del marco, con los soportes de los moto-

res, etc...). Para solucionar esto, se ha reducido el espacio de coordenadas del dispositivo a (-26, 26) para el eje X y (-35, -2) para el eje Y, considerando todos los valores fuera de estos rangos iguales al máximo o mínimo del rango que corresponda.

2.4. Generación de Melodías y Algoritmo de Control

Para la generación inteligente y controlada de melodías, se propone un enfoque híbrido en el que el algoritmo que genera la melodía tiene en cuenta tanto la cadena de Markov como los valores de referencia calculados en función de la posición del dispositivo. La figura 5 resume las diferentes etapas relacionadas con el algoritmo de control y la generación de melodías que se detallan a lo largo de esta subsección.

El componente principal de la composición de las melodías es la cadena de Markov, ya que es la que determina cuáles son las transiciones posibles para cada estado (nunca sonará una transición que no esté presente en el modelo) y las probabilidades iniciales de cada una de estas transiciones. A continuación, se calculan la nota y duración de referencia dada la posición del dispositivo en los ejes Y y X respectivamente y se modifican las probabilidades "teóricas" de la nota siguiente en función de esta referencia. Esta modificación se realiza de dos formas complementarias: por un lado se aumenta la probabilidad de las transiciones más cercanas al valor de referencia, y por otro se evitan todas aquellas transiciones demasiado alejadas de este valor.

La probabilidad de las transiciones cercanas al control se aumenta en función de un parámetro "k" que determina el peso de la referencia a la hora de calcular la probabilidad de cada transición t según la formula siguiente:

*P(t) = k * P(t según referencia) + (1-k) * P(t según cadena de Markov)*

La probabilidad de una transición según el control se calcula en función de la distancia entra esa transición y la transición de referencia. Como cada transición se caracteriza por una nota y una duración, existen dos distancias para cada transición: la distancia de la nota y la distancia de la duración. La distancia se ha definido como el valor absoluto entre la nota de referencia (o la duración de referencia) y la nota (o duración) de una determinad transición.

Para una transición exactamente igual que la de referencia, se ha decidido que la probabilidad según la referencia es 0.4. Para todas las transiciones posibles igual a un único valor de referencia, nota o duración, y cuya distancia respecto al valor que no comparten es 1 (nota o duración anexa), la probabilidad de referencia se ha determinado como 0.1. Por último, si la distancia tanto de la nota como de la duración es 1 (nota y duración anexas a los valores de referencia), la probabilidad de referencia se ha establecido a 0.05. De este modo, si existen todas las transiciones anexas a la transición de referencia, la suma de las probabilidades de referencia sería igual a 1.

Por otro lado, también se eliminan las transiciones que se alejan demasiado de los valores de control. Esta limitación del rango de salida se realiza mediante dos parámetros ajustables que determinan la distancia máxima de las notas y de la duración de las transiciones de salida respecto a los valores de referencia. Las

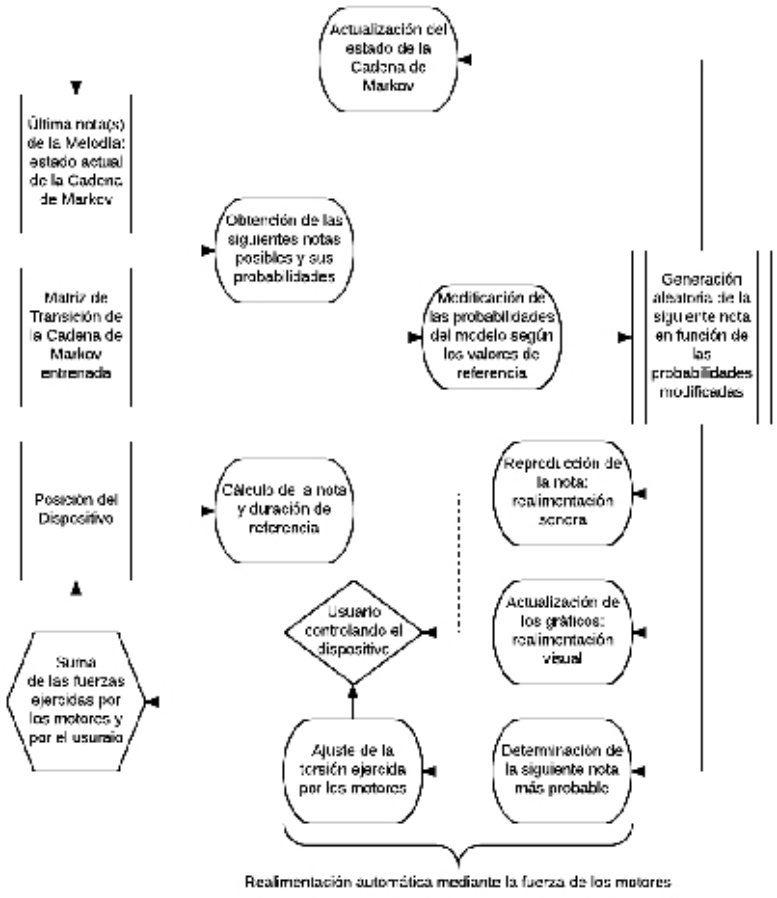


Figura 5: Esquema del algoritmo de control

distancias utilizadas son las mismas que las explicadas en el párrafo anterior, y simplemente se eliminan todas aquellas transiciones cuya distancia al control es superior a la especificada en el parámetro correspondiente. Existe una única excepción a esta regla: los silencios siempre se consideran dentro del rango de control de las notas (no de la duración), para evitar modificar innecesariamente el ritmo de la melodía generada.

Estas modificaciones pueden dar lugar a casos en los que la suma de las probabilidades de las transiciones restantes sea inferior a 1, pero esto se ha tenido en cuenta para el cálculo de la transición siguiente en función de estas probabilidades. Nótese que en el caso de que ninguna transición se encuentre dentro del rango de control (es decir, que todas las transiciones posibles estén

demasiado alejadas del punto de control) se utilizan las probabilidades originales de la cadena de Markov para evitar que la generación de melodía se pare.

Por último, para determinar la siguiente nota de la melodía, se tratan las probabilidades de las transiciones restantes como una distribución estadística y se calcula la nota de salida generando una observación "aleatoria" en función esta distribución.

También se ha a'nadido otra particularidad al control: la posibilidad de acortar notas demasiado largas desde el punto de vista del usuario. La lógica detrás de esto es muy sencilla: si se ha generado un nota larga y se desplaza el *joystick* hacia la izquierda en el eje X (hacia el valor 0, es decir hacia notas más cortas), en el momento en el que la duración indicada por la referencia sea inferior a la duración transcurrida desde el inicio de la nota, la nota actual se "corta" y se adelanta el cálculo de la siguiente.

2.5. Visualización de la Melodía y Realimentación

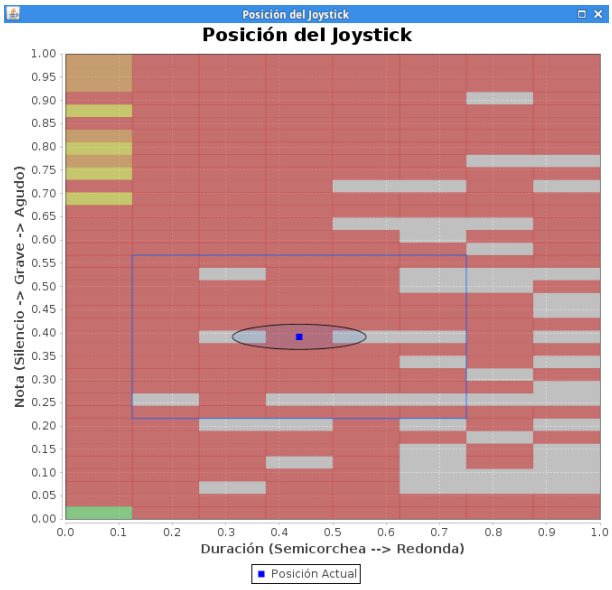


Figura 6: Posición del dispositivo

Para facilitar la visualización de la generación de la melodía se han construído dos gráficos dinámicos. El primero, que se puede ver en la figura 6, muestra la posición actual del *joystick* en un espacio normalizado entre 0 y 1. El fondo de este gráfico se rellena de forma dinámica, mostrando todas las transiciones posibles en función del estado actual de la cadena de Markov y coloreándolas de rojo, naranja, amarillo o verde en función de su menor o mayor probabilidad teórica

(sin tener en cuenta la posición del dispositivo). De este modo el usuario puede hacerse una idea de un vistazo sobre las transiciones posibles, y sus respectivas probabilidades, lo que le permite realizar un control más pertinente. También se dibuja una elipse sobre el espacio que corresponde a aquellas transiciones cuya probabilidad se aumenta por el control, y se delimita mediante un rectángulo azul el espacio en el que entran las transiciones permitidas por el control (en función de las distancias máximas explicadas en la subsección 2.4). El objetivo es, junto al fondo con las transiciones coloreadas, que el usuario pueda saber fácilmente si el control va a afectar o no a la melodía y si está dirigiéndola hacia espacios más o menos probables. Además de servir de ayuda para la visualización del proceso, este gráfico también se puede utilizar para dirigir la generación de la melodía cuando no está conectado el dispositivo.

Por su lado, el segundo gráfico (ver figuras 7, 8, 9 y 10) muestra un histograma con las notas y la duración de referencia en función del tiempo, utilizando como unidad el tick de la secuencia MIDI (eje horizontal del gráfico). El histograma de las notas de referencia se representa mediante la línea roja en la escala normalizada MIDI (valor entre -1 y 35, siendo -1 el valor que representa los silencios). El histograma de las duraciones de referencia se representa a su vez mediante la línea azul, utilizando el número de semicorcheas de duración como escala. A medida que se va generando la melodía, se va representando en tiempo real en el fondo de este gráfico mediante una serie de rectángulos amarillos que representan las notas generadas. La posición del rectángulo en el eje vertical indica la nota (según el mismo eje que para el histograma de la nota de referencia), y su anchura se corresponde con su duración en semicorcheas (en este proyecto, la resolución MIDI está configurada de forma que un tick corresponde con una semicorchea). Si una nota se corta manualmente de la forma explicada en 2.4, este evento se representa mediante una barra vertical roja sobre el rectángulo de la nota que se ha cortado. El objetivo de este segundo gráfico es facilitar la visualización de la melodía generada y el impacto de la referencia sobre su evolución.

También se propone una realimentación mediante la fuerza de los motores del dispositivo. El objetivo es que el dispositivo tienda a desplazarse automáticamente hacia el punto correspondiente con la la transición (nota y duración) más probable en función el estado actual. Para ello, se calcula un vector de fuerza en el espacio (x, y) de forma proporcional a la distancia entre la posición actual del dispositivo y la posición correspondiente a esta transición objetivo. Se trata de un algoritmo de control proporcional, por lo que también es necesario aplicar una constante proporcional directa multiplicando el vector resultante por un factor de "ajuste" que permite que la fuerza de realimentación sea más o menos notable. Valores muy peque'nos de esta constante se traducen en una realimentación insignificante, y valores demasiado grandes generan movimientos demasiado bruscos que provocan que el dispositivo oscile alrededor de la posición objetivo o que incluso uno de los brazos llegue a impactar con el soporte del motor opuesto. Para evitar estas ocurrencias, ya que pueden da'nar el dispositivo, la fuerza de realimentación se ha ajustado a un valor relativamente sutil, y se recomienda que siempre que se manipule el dispositivo no se deje de sujetar hasta que no se

haya finalizado la experimentación. Mediante esta realimentación directamente a través del dispositivo, se ofrece una realimentación "intuitiva" al usuario sobre el control que está realizando: si se deja llevar y desplaza el dispositivo hacia los puntos que menos resistencia presentan, estará generando una melodía más en la línea de lo establecido por el modelo. Sin embargo, si intenta desplazar el dispositivo hacia puntos cuyas transiciones correspondientes sean menos probables, se dará cuenta rápidamente ya que la fuerza que debe ejercer para realizar ese desplazamiento será más importante.

3. Experimentos

3.1. Orden del Modelo de Markov

El primer experimento propuesto consiste en estudiar el impacto del orden de la cadena de Markov sobre la melodía generada. Por este motivo se han ajustado los parámetros relacionados con el control de forma que la melodía no se vea afectada por la posición del dispositivo (peso de la referencia, k, igual a 0 y la distancia máxima a la referencia al mayor rango posible para que no se elimine ninguna transición). También se han elegido unos datos de entrada completamente aleatorios, sobre los que se han ido entrenando modelos de diferente orden. Después de cada entrenamiento, se ha calculado el número de estados diferentes presentes en cada uno y el porcentaje de esos estados para los que existe más de una transición posible (es decir, que se puedan ver afectados por el usuario). Los resultados obtenidos, que se pueden ver en la tabla incluída a continuación, muestran que según aumenta el orden del modelo, el número de estados con más de una transición posible disminuye notablemente lo que afecta del mismo modo a las posibilidades de que la melodía se vea afectada por la referencia. Por ejemplo, si se quisiera utilizar un modelo de cuarto orden con estos datos aleatorios, el usuario únicamente podría afectar a la melodía generada aproximadamente un 15 % del tiempo, siendo el resto del tiempo generada de forma completamente automática. Sin embargo, los modelos de orden superior también parecen generar mejores melodías. Esto se puede explicar por la mayor "memoria" de la que dispone el modelo, al tener en cuenta un mayor número de notas precedentes. Dados estos resultados, el modelo de tercer orden ofrece la solución más equilibrada ya que genera melodías relativamente más elaboradas y además ofrece un número razonable de transiciones controlables (28 %).

Orden del Modelo	Número de Estados	Estados con más de una transición posible
1	296	100 %
2	48 518	68 %
3	306 668	28 %
4	595 106	15 %
6	931 516	6 %
8	1 091 844	4 %

3.2. Entrenando por Estilos

Para un segundo experimento se ha buscado determinar el impacto de los datos de entrada sobre la melodía generada. Se ha ajustado la generación de la melodía de la misma forma que para el experimento previo: evitando cualquier tipo de modificación que dependa de la posición del dispositivo. Dado que este impacto es evidente (la generación de la melodía depende directamente de los datos de entrada), el objetivo de este experimento es ver hasta qué punto se pueden diferenciar diferentes estilos mediante esta metodología. Para ello se ha entrenado un modelo de tercer orden (un valor equilibrado según el experimento anterior) con melodías de tres compositores clásicos: *Johann Sebastian Bach*, *Frédéric Chopin* y el espa'nol *Isaac Albéniz*. Los resultados obtenidos son relativamente ambiguos ya que, pese a que se aprecia una cierta diferencia de estilo, el resultado generado sigue siendo bastante aleatorio. El ritmo parece que se captura mejor pero las variaciones de tono (notas) no están al nivel que se podría esperar.

3.3. Entrenando Por Tonalidad

Teniendo en cuenta los resultados del experimento previo, se ha decidido entrenar el modelo únicamente con las canciones de una determinada tonalidad de un determinado compositor. El objetivo de este tercer experimento es comprobar si con los datos adecuados, se puede generar una melodía con mejores variaciones de tono. El compositor elegido ha sido de nuevo *Johann Sebastian Bach* para poder comparar el resultado con los anteriores en los que no se tenía en cuenta la tonalidad. Tras clasificar las canciones por tonalidades en función de las informaciones obtenidas en [2], se ha entrenado un modelo de Markov de tercer orden con todas aquellas de las tonalidades Sol Mayor y Re Menor. Esta vez, los resultados obtenidos son mejores ya que las melodías generadas son bastante más curiosas: parece que el modelo ha conseguido aprender la tonalidad.

3.4. Parámetros de Control

Por último, se han realizado varios experimentos para estudiar la influencia de los diferentes parámetros relacionados con el control. En este caso se ha entrenado un modelo de segundo orden (por ofrecer un número considerable de transiciones en las que pueda afectar la referencia) con los datos de la tonalidad Sol Mayor utilizados previamente. Se ha procurado realizar un control similar en todos los experimentos para que sean fácilmente comparables.

El primer parámetro estudiado ha sido “k”, el peso de la referencia, variando su valor de 0,3 a 0,8. La distancia máxima entre la referencia y la salida ha sido dejada en unos valores intermedios de 6 para las notas y 2 para las duraciones. Los resultados pueden verse en los histogramas de las figuras 7 y 8. Se puede observar que con un parámetro “k” de 0,3 la melodía generada es más variada, ya que oscila más alrededor de la nota de referencia que en el caso de 0,8, cuando sigue esta misma referencia de forma más rigurosa. Se observa lo mismo para las duraciones, que se muestran relativamente más estables con el valor superior.

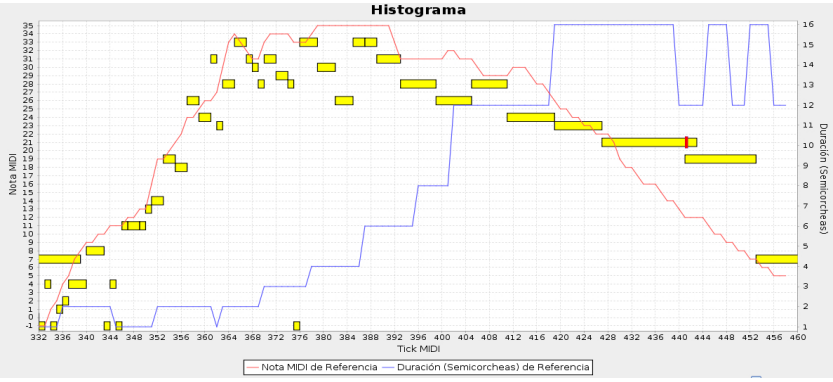


Figura 7: Melodía generada con k = 0.3

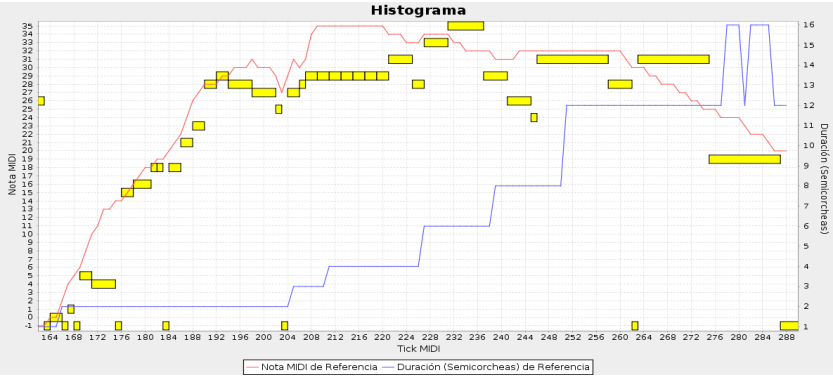


Figura 8: Melodía generada con k = 0.8

También se ha estudiado el impacto del rango de control. Para estos experimentos se ha ajustado el parámetro "k" a 0,5. Por un lado se ha experimentado con un rango estrecho, permitiendo únicamente aquellas transiciones alejadas como mucho de 3 notas y una duración (dentro de las 8 duraciones normalizadas posibles) respecto a la referencia. Por otro, se ha realizado el mismo experimento con un rango más amplio que permite notas hasta a una octava de distancia (12 notas) y a 3 duraciones de la referencia. Los resultados pueden verse en las figuras 9 y 10. El rango estrecho obliga a la melodía a seguir más de cerca a la melodía mientras que el rango más amplio permite una mayor libertad. Nótese que, tal y como se explica en 2.4, si no existe ninguna transición posible dentro del rango permitido se ignora la posición del dispositivo para evitar parar la generación de la melodía.

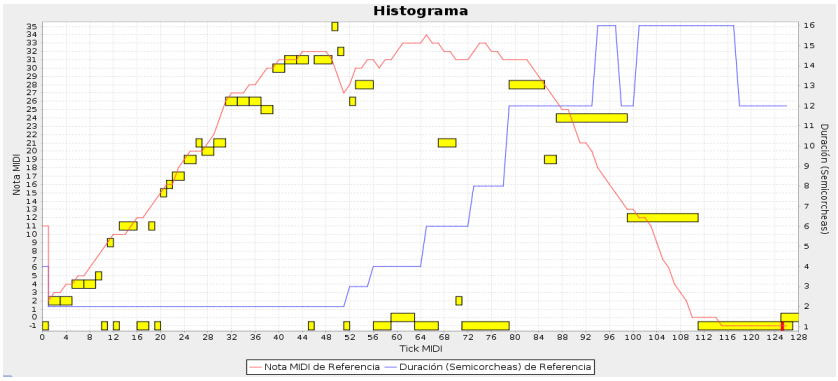


Figura 9: Melodía generada con un rango posible estrecho

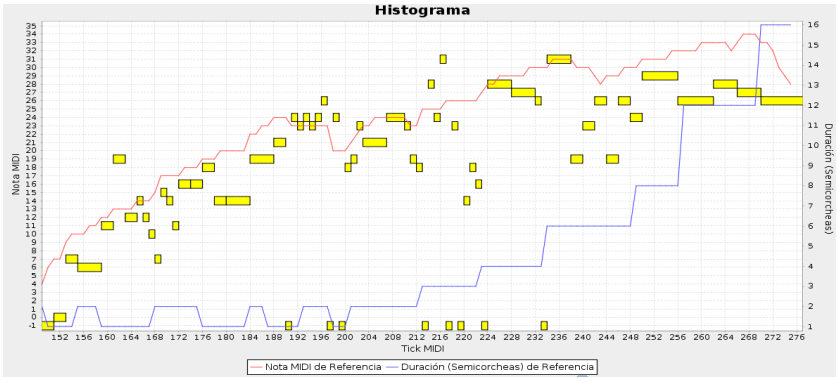


Figura 10: Melodía generada con un rango posible amplio

Gracias a estos experimentos se ha podido apreciar que el parámetro “k” es de mayor utilidad cuando existen muchas transiciones posibles (permite mayor precisión), mientras que delimitar el rango funciona mejor en casos con un menor número de transiciones (permite alcanzar transiciones alejadas más facilmente). Pese a parecer dos parámetros que realizan la misma función a primera vista, utilizados correctamente son complementarios y permiten al usuario ajustar de manera relativamente intuitiva la manera de la que desea “dirigir” la generación de la melodía.

4. Conclusión

A lo largo de este artículo se ha presentado el proceso de investigación realizado para estudiar las posibilidades de la utilización de un dispositivo de tipo

joystick para la composición inteligente controlable de melodías, utilizando los ejes X e Y del dispositivo para el control de la duración y el tono de las notas generadas. El enfoque propuesto utiliza un conjunto de archivos MIDI, de los cuales se extraen las notas con sus respectivas duraciones reagrupadas por grupos de instrumentos. A continuación, se entrena una cadena de Markov con los datos del grupo deseado, y se van modificando las probabilidades de transición de este modelo en función del control para generar una melodía aleatoriamente respetando estas probabilidades "controladas".

Para realizar esta investigación se ha desarrollado una aplicación que podría ser descrita como un secuenciador inteligente controlable, ya que por una parte aprende a generar secuencias a partir de conjuntos de ejemplos y por otra admite la intervención directa del usuario para guiar el proceso de generación de la melodía. La acción del usuario sobre el sistema se ha implementado de dos formas: mediante el movimiento del ratón sobre una ventana interactiva o bien manejando un dispositivo mecánico tipo *joystick* capaz de recibir realimentación del propio modelo a través de sus motores.

Esta aplicación también permite configurar ciertos parámetros de control para modificar el peso relativo de la acción del usuario sobre la secuencia generada por el modelo o la resistencia que el dispositivo ofrece. Aunque el algoritmo de control propuesto se ha particularizado al dispositivo concreto que se presenta en este trabajo, sería fácilmente adaptable a otros tipos de dispositivos o actuadores.

Los resultados de los diferentes experimentos realizados remarcan la importancia de los datos de entrada sobre la melodía generada. Al entrenarse únicamente con dos dimensiones, las notas y sus duraciones, el modelo de Markov desconoce otras características importantes de las melodías extraídas como puedan ser la tonalidad o la armonización. Sin embargo, teniendo este factor en cuenta, se pueden seleccionar únicamente los datos de una tonalidad en particular, por ejemplo, para que la melodía generada respete esta tonalidad y los resultados sean mejores.

Referencias

1. Midi messages specifications. `https://www.midi.org/specifications/item/table-1-summary-of-midi-message`
2. Bach, J.: Obras de bach por tonalidades. `http://www.jsbach.es/`
3. Bergsland, A., Wechsler, R.: Composing interactive dance pieces for the motioncomposer, a device for persons with disabilities. In: Berdahl, E., Allison, J. (eds.) Proceedings of the International Conference on New Interfaces for Musical Expression. pp. 20–23. Louisiana State University, Baton Rouge, Louisiana, USA (May 31 – June 3 2015), `http://www.nime.org/proceedings/2015/nime2015_246.pdf`
4. Fernández, J.D., Vico, F.: Ai methods in algorithmic composition: A comprehensive survey. Journal of Artificial Intelligence Research 48, 513–582 (2013)
5. Google: Deep dream. `https://github.com/google/deepdream`
6. Google: Magenta. `https://github.com/tensorflow/magenta`
7. Hass, J.: How does the midi system work? `http://www.indiana.edu/~emusic/etext/MIDI/chapter3_MIDI4.shtml`

8. Jungleib, S.: General Midi. AR Editions, Inc. (1996)
9. Lorenzo, L.M.I.: Intelligent midi sequencing with hamster control. https://courses.cit.cornell.edu/eceprojectsland/STUDENTPROJ/2002to2003/lil2/hamsterMIDI_done_small.pdf
10. Odersky, M.: The scala language specification
11. Pachet, F., Roy, P.: Markov constraints: steerable generation of markov sequences. Constraints 16(2), 148–172 (2010), http://dx.doi.org/10.1007/s10601-010-9101-4

Sistema de gestión inteligente de préstamo de bicicletas eléctricas

Jorge Revuelta Herrero[1], Juan Manuel Corchado Rodríguez[1]

Departamento de Informática y Automática, Universidad de Salamanca.
Plaza de la Merced s/n. 37008, Salamanca, España
{jrevuelta,corchado}@usal.es

Resumen En la actualidad se pueden encontrar numerosos servicios urbanos, tanto en territorio nacional como internacional, relacionados con el alquiler y disposición de vehículo al uso de los suscriptores del servicio. La normalidad de esta clase de servicio de alquiler de transporte urbano está centralizada en la bicicleta, ya que conforma un vehículo personal, de costes bajos y con nula huella medioambiental.
En este trabajo se presenta un método de optimización que busca minimizar el tiempo en que los usuarios de un servicio de alquiler de bicicletas deben esperar para realizar sus viajes a través de la adaptación de los lugares en que los usuarios deben depositar sus bicicletas al finalizar sus recorridos.

Keywords: Bicicletas, BSS, Bicicleta Eléctrica

1. Introducción

El principal problema al que se enfrentan los sistemas de alquiler de bicicletas es la satisfacción de la mayor cantidad de demandas realizadas por parte de los usuarios de la plataforma.

En la literatura, la principal línea en la que se aborda este problema es la utilización de una flota de vehículos de transporte de determinado tamaño que, a lo largo de la jornada, se encarga de redistribuir los recursos entre los diferentes puntos de alquiler que ofrece el servicio.

Este trabajo se centra en la creación de un método de optimización que permita minimizar los tiempos de espera de los usuarios de una plataforma de alquiler de bicicletas alterando de manera controlada parámetros de una serie de viajes conocidos para unos usuarios a lo largo de una jornada. El método se basa en la implementación de dos algoritmos individuales: una heurística que permite alterar el lugar de destino de los viajes realizados por los usuarios y una segunda heurística que permite alterar tanto el lugar de origen como de destino de los viajes realizados por los usuarios. Realizar alteraciones que suponga un desplazamiento no previsto para los usuarios penaliza el rendimiento de la solución, que siempre debe ser el mínimo posible pudiendo satisfacer sin un tiempo elevado de espera todas las demandas de los usuarios.

La sección 2 se realizará una revisión de trabajos previos relacionados con la optimización en el campo de los servicios de alquiler de bicicletas. La sección 3, presenta los métodos propuestos, las bases teóricas que lo sustentan y la metodología que llevan a cabo. En la sección 4 se realiza la aplicación de los métodos propuestos sobre un problema concreto. En la sección 5 se presentan los resultados de la aplicación de los métodos propuestos sobre el caso de estudio planteado y se obtienen las conclusiones pertinentes.

2. Estado del arte

En la literatura existe numerosos trabajos y artículos relacionados con problemas de optimización en el campo de los sistemas de alquiler de bicicletas. Cada uno de ellos aborda el mismo problema a resolver: optimizar de la manera más eficiente los recursos del servicio para que los usuarios hagan uso del mismo con los menores contratiempos posibles.

El trabajo propuesto por [1] plantea un problema relacionado con los *Urban Bicycles Renting Systems*, donde su infraestructura requiere de la optimización de las rutas de vehículos que conectan las diferentes paradas del servicio con los centros de almacenamiento. En el estudio, se plantea resolver el problema como un *VRP* con múltiples almacenes, al cual se le aplican técnicas inspiradas en la naturaleza, en concreto una fusión de algoritmos evolucionarios [2] con *Ant Colony Systems* [3].

En el trabajo propuesto por [4] se aborda el problema de sistemas de alquile de bicicletas donde el factor crucial a optimizar es la capacidad del usuario de poder recoger la bicicleta desde el sitio que elige como punto de origen y poder depositarla en la parada más cercana a su destino. Para ello, los autores aplican sistemas de balanceo en un problema *VRP* considerando un caso estático. Los vehículos llevan a cabo rutas por las estaciones y deben ser devueltas a localizaciones concretas que son conocidas a priori, y todas y cada una de las estaciones debe ser visitada una única vez y una sola vez por vehículo. Se aborda el problema como un problema *traveling salesman* [5] con restricciones adicionales.

En el trabajo de [6] proponen un trabajo relacionado con el rebalanceo de sistemas de alquiler de bicicletas a través de un algoritmo de destrucción y reparación. Se realiza la combinación de una metaheurística de destrucción y reparación que hace uso de una heurística añadida constructiva y de varios procedimientos de búsuqeda local. El algoritmo propuesto es adaptado para resolver el *one-commodity Pickup and Delivery Vehicle Routing Problem with maximum Duration (1-PDVRPD)*, que es la variante del *Bike Sharing Problem* donde una duración máxima para cada ruta es la restricción principal.

Otros trabajos, sin embargo, como el propuesto por [7], centran la solución de la falta de recursos en las demandas de los usuarios en los sistemas de alquiler de bicicletas, en la construcción de las estaciones a partir de una serie de recursos máximos disponibles, optimizando sus localizaciones en función de las demandas por áreas geográficas de un área total a cubrir por el sistema de alquiler. El

método propuesto hace uso de un método de optimización para el diseño de un sistema de alquiler de bicicletas tal que maximice la demanda cubierta y utilizando el presupuesto como restricción. Combina decisiones estratégicas para la localización de paradas de bicicletas y definiendo la dimensión del sistema (número de estaciones y bicicletas) con decisiones operativas (relocalización de bicicletas). El modelo final determina la óptima localización de las estaciones, el tamaño de la flota, la capacidad de las estaciones y el número de bicicletas por estación.

En el artículo propuesto por [8] divide el problema relacionado con los sistemas de alquiler de bicicletas en dos fases principales: la primera estima la demanda insatisfecha en cada una de las estaciones por un período de tiempo en el futuro y por cada número posible de bicicletas al comienzo de cada período; la segunda fase utiliza estas estimaciones para guiar los algoritmos de redistribución propuestos. El parámetro de calidad de servicio propuesto utiliza datos conocidos que provee el sistema de alquiler y realiza aproximaciones y predicciones estadísticas para tiempos futuros para cada parada dado un número de bicicletas inicial en el establecimiento. Esta información de medida de calidad es aplicada en la segunda fase, un *VRP* en el que se ajustan las rutas de los vehículos que relocalizan las bicicletas en las estaciones con mayor demanda y recoge las bicicletas dañadas para devolverlas al almacén.

3. Método propuesto

En esta sección se describen los métodos y algoritmos propuestos según las limitaciones y los diferentes puntos de vista a la hora de abordar el problema que se han visto en la sección anterior.

El principal impedimento a la hora de implementar el método que se propone es el hecho de que la búsqueda en el espacio de la solución es discreta. Solo las coordenadas del espacio de búsqueda que se corresponden con paradas en los diferentes instantes temporales (que también son discretos), son posibles soluciones o pasos para la resolución del problema.

De la misma forma, la componente de tiempo que añade una dimensión más al problema hace que la aplicación de alguno de los algoritmos presentados anteriormente queden descartados. Las dependencias entre las diferentes soluciones realizadas para cada conjunto de viajes se propagan en el tiempo de manera encadenada hasta la obtención de la solución final.

Es por esto, que la heurística propuesta por el método se debe de implementar de forma iterativa, y de tal forma que la solución al problema no sea perteneciente a una única instancia del algoritmo (solución local) si no a un conjunto finito de ellas que sean resueltas en paralelo y que sean capaces de obtener la mejor solución entre todas ellas como solución global del problema.

El método utilizado puede ser semejante al aplicado por una búsqueda *Scatter* [9], [10], [11], ya que las soluciones pertenecen al dominio de punto discretos del espacio de búsqueda (las coordenadas de las paradas del servicio) junto con una implementación de búsqueda en espacio discreto semejante a la propuesta

en *Swarm Particle Optimization* [12], en el que es el conjunto de partículas el que busca individualmente las soluciones candidatas del problema y no el propio conocimiento individual del enjambre de cada una donde se encuentra la solución al problema.

La idea básica del procedimiento parte de optimizar que un conjunto de usuarios que realizan unas rutas concretas a unas horas de partida que ellos predeterminan, puedan realizarlas con el menor tiempo de transporte total posible, y que aun con la ejecución de todas las rutas, se eviten situaciones en las que un usuario no encuentre una bicicleta en el punto de origen en el momento de realizar su ruta.

El sistema puede ser modelado en forma de grafo totalmente conectado, en el que cada uno de los nodos es una parada de bicicletas y los enlaces las rutas posibles que pueden llevar a cabo los usuarios.

Cada uno de los nodos del grafo, que representa una parada de bicicletas en nuestro sistema, contiene información sobre su estado, así como la inclusión de estados derivados de la demanda de los usuarios al realizar rutas. Es por esta razón que a los nodos se les incluye una tabla de recursos y demandas que permite gestionar al sistema las capacidades de cada una de las paradas de manera local para obtener la mejor solución de manera global.

La forma en que los recursos y las demandas se representan siempre van relacionadas con una marca temporal que permite al sistema situar en unidades de tiempo discretas la llegada y salida tanto de recursos como de demandas.

Figura 1: Ejemplo de tablas de recursos y demandas para una parada en el sistema.

Tal y como se puede observar en la figura 1, cada una de las paradas se ve definida en un punto inicial de la jornada de viajes a través de la demanda que tiene cada usuario de la plataforma de viajar desde un punto concreto hasta un punto final a una hora concreta del comienzo de la ruta. Además, las paradas tienen bicicletas disponibles a partir de un momento temporal para ser utilizadas por los usuarios de la plataforma.

Es fácil, por tanto, que nos encontremos en la situación de que un usuario que quiera realizar una ruta en un momento dado, se encuentre que no existen bicicletas en su puesto de origen. Es, por tanto, el objetivo del método propuesto minimizar el tiempo no útil de todos los usuarios que realizan rutas con bicicletas,

entendiendo tiempo no útil como la suma de los tiempos de espera de los usuarios que no encuentran una bicicleta en su puesto de origen en el momento de la demanda como el tiempo que un usuario invierte en viajar de la parada que el sistema le recomienda a su origen o destino escogidos.

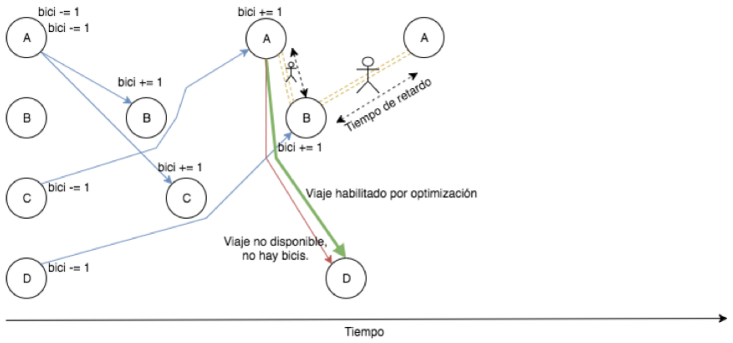


Figura 2: Representación gráfica del método de optimización propuesto.

Como se puede observar en la ilustración 2, el método propuesto permite al sistema enviar a los usuarios a paradas que no coinciden con las preferencias escogidas por el usuario en su ruta, permitiendo así que el sistema pueda reponer de bicicletas otras paradas que poseen más demandas y minimizando los tiempos globales de espera. Este desplazamiento producido por el ajuste del método tiene en cuenta que el usuario debe viajar por defecto de manera más lenta entre la preferencia del usuario y la recomendación del sistema, por lo que ese coste en tiempo también es utilizado como medida a minimizar en el algoritmo.

El fundamento del algoritmo se basa en un espacio tridimensional donde se encuentran los planos bidimensionales espaciales *(x,y)* paralelos a la tercera dimensión *(t)*, que representan las localizaciones geográficas de las paradas en nuestro problema. Cada demanda de un usuario genera un plano de petición de servicio coincidiendo con la demanda de viaje por un usuario y pretende encontrar un P' para cada P resultado de la realización del viaje y que suma los sucesivos retardos ocasionados de manera iterativa para cada demanda.

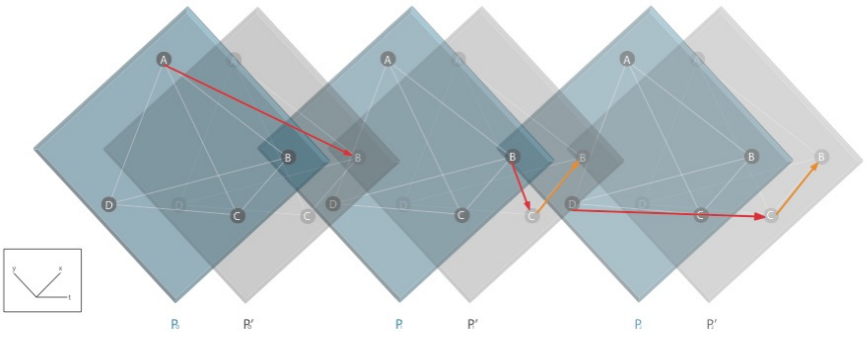


Figura 3: Representación gráfica tridimensional del método de optimización propuesto.

El resultado del algoritmo es el conjunto de planos de demanda $P_0, ..., P_n$ y planos destino $P'_0, ..., P'_n$ que representan los destinos escogidos y que suman todos los retardos ocasionados T_d.

El primero de los métodos propuesto contempla únicamente la modificación del destino al que se envía al usuario para cada una de las demandas del sistema y se muestra a continuación:

1. Se genera un espacio $2D$ de tamaño $m * n$ en el que se sitúan de manera aletoria k paradas con b bicicletas iniciales cada una.

$$\begin{pmatrix} (k_i|0)_{0,0} & \cdots & (k_i|0)_{0,n} \\ \vdots & \ddots & \vdots \\ (k_i|0)_{m,0} & \cdots & (k_i|0)_{m,n} \end{pmatrix}, k_i \in K = k_0, ..., k_k \tag{1}$$

2. Se genera un vector de tamaño definido con d demandas en coordenadas (x, y) correspondientes a alguna de las $k_n \in K$ paradas.

$$D = d_0, ..., d_n, \qquad d_i = (x_i, y_i), t_i \tag{2}$$

3. Se genera además, un instante discreto t para cada $d \in D$ que representa el instante discreto en el que se produce la demanda del viaje.

$$\forall d_i \in D \, \exists \, t_i \in \mathbb{N} \tag{3}$$

4. El algoritmo por tanto debe optimizar que se produzca la menor cantidad de tiempo de retardo en las demandas en D teniendo en cuenta las limitaciones de b_k bicicletas disponibles en las k paradas y las modificaciones que éstas sufren en el tiempo debido a la ejecución de las demandas en D.

5. Para cada demanda $d_i \in D$:
6. Escoger el plano $(P|P')_k$ más cercano temporalmente anterior al plano P_i como estado de origen o escoger el plano de estado inicial P_0 si no existe otro.
7. Si es posible viajar desde el origen de $d_i \exists\ k_d \mid b_d > 0$ esto es, bicicletas disponibles, se utiliza como punto de origen y se substrae una bicicleta de k_d.
8. Se genera una coordenada de desviación aleatoria sobre la coordenada de destino ideal para la demanda, siguiendo una distribución gaussiana de centro las componentes x, y de la coordenada de destino y desviación configurable, y se obtiene la parada más cercana a la coordenada generada.
   En el caso de que la parada obtenida tenga recursos (localizaciones de aparcamiento disponibles) suficientes, se utiliza como destino y, en el caso de que no los hubiera, se amplía la desviación estándar configurada en una unidad de la distribución gaussiana y se obtiene otra coordenada.
   Este paso es repetido hasta que exista una parada con recursos suficientes diferente a la de origen de la demanda o hasta que se realizan un número máximo de u iteraciones.
   En caso de no haber obtenido solución en la búsqueda de destino, la solución planteada se descarta como solución candidata.

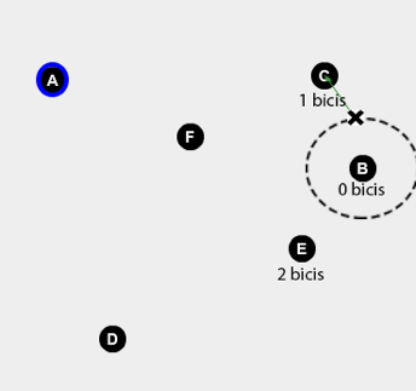


Figura 4: Generación aleatoria de localizaciones por rangos de desviación estándar.

9. Para la parada de destino escogida, se añade el recurso depositado (la bicicleta que se deja en la parada) y se comprueba que el destino coincide con el escogido por el usuario. En caso de no coincidir se obtiene el tiempo de viaje (determinado por la distancia de Manhattan en la matriz de localizaciones) desde el destino seleccionado al destino deseado multiplicado por el factor de diferencia de velocidad de viajar en bicicleta respecto a pie y se le suma

al T_d o *delay time* global del problema.

$$\begin{aligned} T_d &= T_d + Tv_{fd_i,fe_i} * k_{peaton-bicicleta} \\ \text{donde} fe_i &= \text{final escogido para } d_i \\ \text{donde} fd_i &= \text{final ideal para } d_i \end{aligned} \tag{4}$$

10. Se calcula el tiempo que se tarda utilizando la distancia de Manhattan del origen al destino seleccionado y se sitúa el plano P_i' en el instante temporal respecto del origen P_i más el tiempo de viaje calculado.
11. Una vez realizados los pasos 5...10 para cada d_i se obtiene el valor T_d final a minimizar.

Además, se propone una variante del método propuesto anteriormente que sí que permite ajustar la localización de origen de las demandas de un conjunto de viajes de los usuarios.

Los pasos que lo componen son los siguientes:

1. Se aplican de manera idéntica los pasos 1 a 4 del método anterior.
2. En el bucle principal que recorre $d_i \in D$, se genera en primer lugar una coordenada aleatoria, siguiendo las mismas reglas que las usadas para generar la de destino. La restricción aplicada en este caso es, que existan bicicletas suficientes en la parada escogida como origen.
3. Se suma la diferencia aplicada en el desplazamiento por la diferencia de utilizar una parada de origen diferente a la definida por el usuario, como se realiza en el paso 10 del método anterior.
4. Se realizan los pasos restantes de manera idéntica al método anterior.

4. Caso de estudio

Con el fin de validar los métodos propuestos anteriormente, se desarrolla un caso de estudio simulado con datos verosímiles que podrían corresponderse con la realidad.

Variables:

- Número de viajes: 50, 100, 150.
- Período de tiempo en el que se producen: 12 horas.
- Número de paradas: 12
- Bicicletas iniciales por parada: 7
- Bicicletas máximas por parada: 12
- Número de partículas por problema: 100

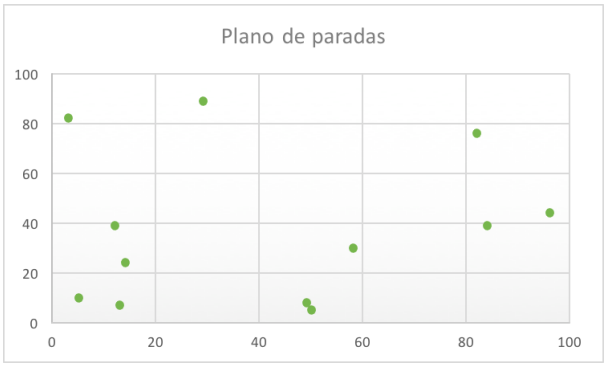


Figura 5: Localizaciones de las paradas para el caso de estudio.

La implementación de los métodos se ha realizado utilizando el lenguaje de programación *Python* en su versión 2.7 utilizando un plano de dimensiones $100x100$ como el que podemos ver en la figura 5 sobre el que se calculan distancias de *Manhattan* (por su similitud a las manzanas en una ciudad).

5. Resultados y conlusiones

El primer caso a poner a prueba para comprobar la validez de los métodos propuestos es la generación de viajes en círculo. Esto es, generar un número de viajes igual al número de bicicletas disponibles con destino la misma parada y desde esa parada generar el mismo número de viajes, así recorriendo todas las paradas y terminando por la que se comenzó. El resultado óptimo es que para este tipo de disposición de viajes, no se deberían producir retardos acumulados en ninguno de los casos, puesto que siempre habrá recursos suficientes y cualquier alteración que provocara desplazamientos a pie, penalizaría el resultado.

Los resultados ofrecen un resultado satisfactorio, obteniendo un $T_d = 0$ en todos los casos.

Además, para poner a prueba los dos métodos propuestos se han aplicado con el mismo conjunto de datos por ejecución y en diferentes cantidades de viajes solicitados, en un rango de 50 viajes hasta 150 viajes.

La generación de los datos para la ejecución de las diferentes pruebas mostradas a continuación es la generación de N viajes dentro del rango de tiempo disponible en segundos (por tanto, para este caso: $[0, 43200]$) con origen aleatorio entre el conjunto de paradas descrito en la sección anterior y destino aleatorio diferente del origen entre todas las paradas del conjunto definido y que es idéntico para todas las pruebas ejecutadas.

Se han realizado 5 pares de pruebas para cada conjunto de viajes de mismo número y se han usado los mismos datos aleatorios generados para cada par, siendo el par compuesto por la mejor solución para los datos del algoritmo *Destination* y la mejor solución para los datos del algoritmo *Origin-Destination*.

También, cabe destacar que se han descartado los pares de pruebas en aquellos casos que cualquiera de los dos algoritmos no ofreciera solución (es decir, solución infinita).

Como se puede observar en las figuras 6, 7 y 8, el método que contempla la modificación del destino de los viajes de las diferentes demandas obtiene en todos los casos mejores resultados de optimización respecto al método que permite modificar también el punto de partida de las demandas en función de las necesidades del sistema.

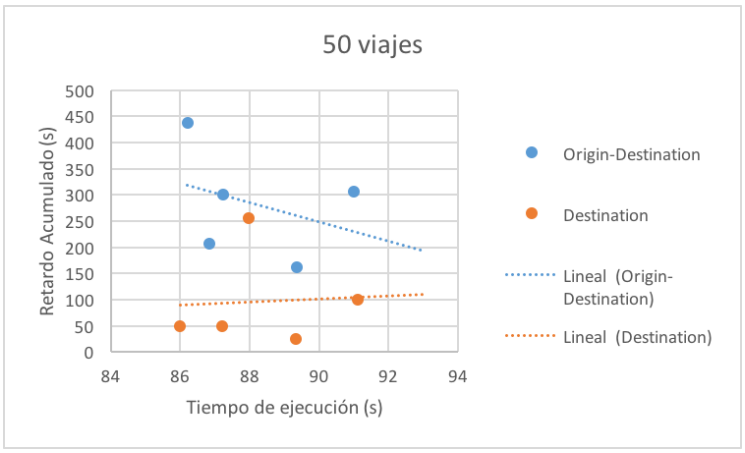


Figura 6: Resultados para 50 viajes.

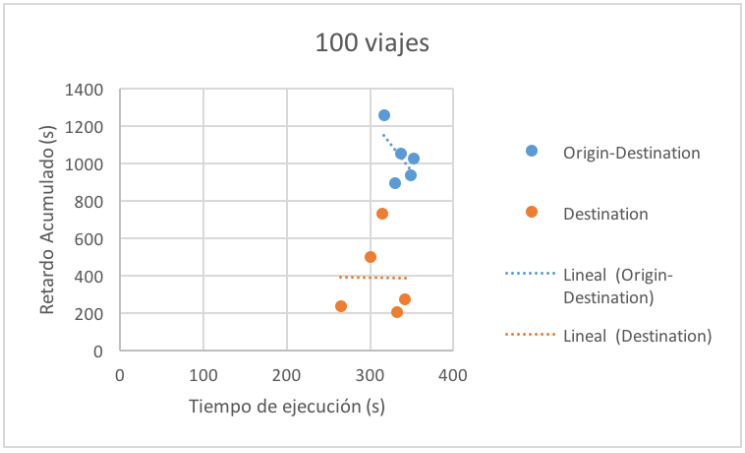


Figura 7: Resultados para 100 viajes.

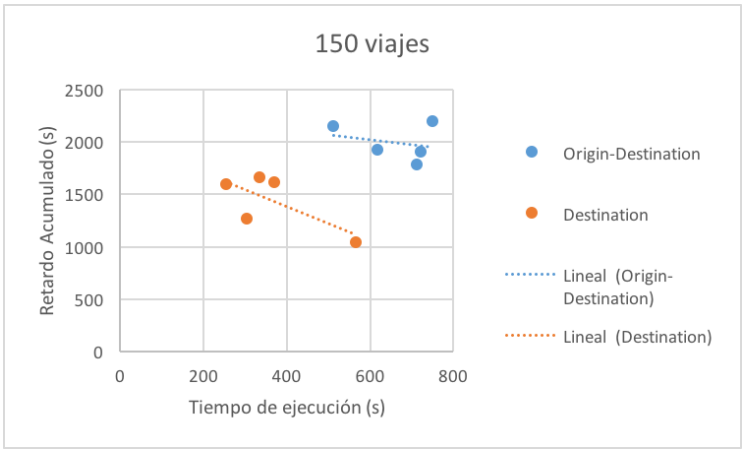


Figura 8: Resultados para 150 viajes.

A su vez, la tabla 1 muestra los resultados de la realización de la prueba T-Test de Student, que nos permite determinar si existe una diferencia significativa entre las medias de los valores de rendimiento calculados por los grupos que conforman la ejecución de cada una de las dos variantes del método propuesto, a través de la probabilidad de que dos resultados de los diferentes grupos sean idénticos.

La hipótesis nula H_0 sería, por tanto, afirmar que los resultados arrojados para cada número de viajes y para los dos métodos propuestos es similar. Puesto que los valores obtenidos son todos $< 0{,}05$ se puede rechazar la hipótesis nula H_0 y afirmar que las dos muestras son significativamente diferentes y que, por lo tanto, los resultados arrojados por los dos métodos difieren entre sí (tomar como cierta H_1).

Tabla 1: Resultados de T-Test para el rendimiento en diferentes conjuntos de viajes.

Viajes	50	100	150
T-Test	0,0186	0,0006	0.0047

Sí que cabe observar que a medida que se aumenta el número de viajes, el método *Destination* puede no encontrar una solución válida (es decir, solución infinita) para el problema debido a las restricciones producidas por la falta de recursos (bicicletas disponibles en las paradas), mientras que el método *Origin-Destination* arroja resultados (aunque con un rendimiento menor) en todos los problemas generados.

Otro aspecto a tener en cuenta es el tiempo de ejecución de ambos métodos, para un número de viajes reducido (que no supera el número de recursos globales del sistema) ambos métodos concluyen en una solución en márgenes de tiempo muy cercanos. Esto no se respeta cuando los recursos del problema comienzan a ser inferiores al numero de demandas en el problema, donde el método Destination obtiene resultados (cuando es capaz de obtenerlos) en menor tiempo que el método Origin-Destination como se muestra en la figura 9.

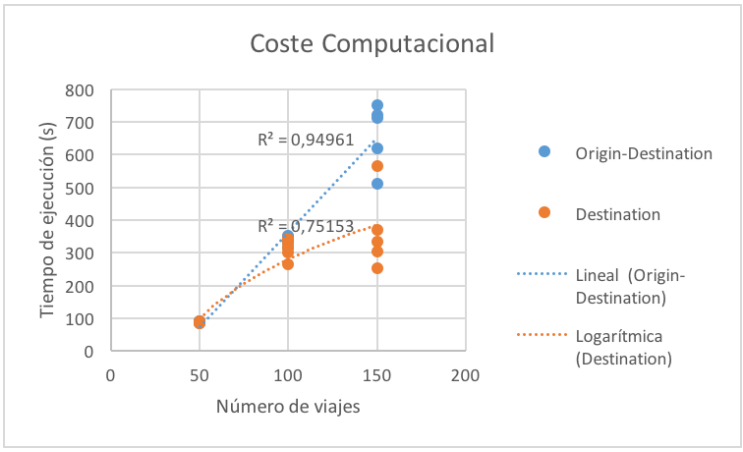


Figura 9: Coste computacional para las ejecuciones.

Uno de los primeros objetivos para el desarrollo futuro de los métodos es ampliar el conocimiento compartido que existe en el método, tal y como se aplica en *Ant Colony Optimization*, cuyas hormigas además de poseer información individual privada, comparten información sobre el conocimiento general adquirido y generar soluciones candidatas válidas a partir de las soluciones locales mínimas.

Referencias

1. C. Chira, J. Sedano, J. R. Villar, M. Cámara, and E. Corchado, "Urban bicycles renting systems: Modelling and optimization using nature-inspired search methods," 2014.
2. T. Bäck, D. Fogel, and Z. Michalewicz, "Handbook of evolutionary computation," *Release*, 1997.
3. M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, vol. 2, pp. 1470–1477, IEEE, 1999.
4. A. A. Kadri, I. Kacem, and K. Labadi, "A branch-and-bound algorithm for solving the static rebalancing problem in bicycle-sharing systems," *Computers & Industrial Engineering*, vol. 95, pp. 41–52, 2016.

5. K. L. Hoffman, M. Padberg, and G. Rinaldi, "Traveling Salesman Problem," in *Encyclopedia of Operations Research and Management Science*, pp. 1573–1578, Boston, MA: Springer US, 2013.
6. "A destroy and repair algorithm for the Bike sharing Rebalancing Problem," *Computers & Operations Research*, vol. 71, pp. 149–162, 2016.
7. I. Frade and A. Ribeiro, "Bike-sharing stations: A maximal covering location approach," *Transportation Research Part A: Policy and Practice*, vol. 82, pp. 216–227, 2015.
8. R. Alvarez-Valdes, J. M. Belenguer, E. Benavent, J. D. Bermudez, F. Muñoz, E. Vercher, and F. Verdejo, "Optimizing the level of service quality of a bike-sharing system," *Omega*, vol. 62, pp. 163–175, 2016.
9. F. Glover, "Heuristics for integer programming using surrogate constraints," *Decision Sciences*, 1977.
10. F. Glover, "Tabu search-part I," *ORSA Journal on computing*, 1989.
11. F. Glover, "Tabu search—part II," *ORSA Journal on computing*, 1990.
12. J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, 1995.

La robótica en la Educación

Domingo Sánchez y M. Angélica González Arrieta

Universidad de Salamanca
Departamento de Informática y Automática
Facultad de Ciencias
Plaza de los Caídos s/n
37008 Salamanca, Spain

domits@usal.es, angelica@usal.es

Resumen El objetivo de la enseñanza de la robótica es lograr una adaptación de los alumnos a los procesos productivos actuales, donde la automatización juega un papel muy importante. Pero también es una disciplina que puede cumplir otros objetivos más allá de los estrictamente laborales (formación complementaria, fomento del trabajo en equipo, I+D+i empresarial, atención a personas con necesidades educativas especiales, mejora del rendimiento en alumnos con elevadas capacidades intelectuales, alumnos con TDAH, ocio...). Incluso en Economía se está imponiendo el término Fintech (Finance and Technology), que aglutina a todas aquellas empresas de servicios financieros que utilizan la última tecnología existente para poder ofrecer productos y servicios financieros innovadores. La robótica se centra en el diseño, planificación, montaje, programación, prueba y rediseño de un robot. Estos robots pueden emplearse en educación, seguridad, tecnología del automóvil, sistemas productivos, exploración espacial, entornos nocivos de la industria química y nuclear, desactivación de explosivos, domótica, intervenciones quirúrgicas, rehabilitación y fabricación de prótesis, prestación de servicios...Este trabajo analiza la importancia que el sistema educativo español confiere a la robótica y a aquellas otras materias de ámbito tecnológico. También se estudia la evolución de esta disciplina, metodología educativa, partes de los robots (sensores, actuadores, programación), diseño e impresión 3D, controladoras, animación y simulación robóticas. Se facilitan prácticas de programación, diseño imprimible de dron y animación 3D. Además, se destaca la importancia que las redes neuronales profundas y la realidad aumentada están teniendo en la robótica. Son áreas de investigación que aportan avances, y lo seguirán haciendo en el futuro, para que la interaccion con el entorno y la operatividad del usuario sean cada vez más eficientes.

Keywords: robótica, educación

1. Introducción

Los avances de la tecnología en los últimos años, en especial en lo relativo al control automático y robótica, han provocado que los sistemas educativos de

todo el mundo enfoquen su mirada hacia estos ámbitos del conocimiento, ya que su estudio permite no sólo un acercamiento al entorno en el cual el alumnado desarrolla su vida, sino también al ambiente altamente tecnificado en el que deberá desarrollar su actividad profesional en el futuro.

Esta materia permite resolver un problema tecnológico real, abarcando el conjunto de actividades pedagógicas dirigidas al diseño, la fabricación y montaje de un robot, las cuales se complementan con la elaboración del programa informatico que permita el control del mismo. También se emplea cada vez más la impresora 3D, un tipo específico de robot que cumple una función esencial dentro de la cultura maker y la filosofía do it yourself (DIY), que engloba procesos de inteligencia, así como de creación colectiva a través de la compartición de códigos, prototipos y modelados.

La programación como herramienta de control es utilizada en numerosos campos técnicos y sistemas de información, y es necesario conocerla para poder comprender y controlar la tecnología que nos rodea. Saber programar es fundamental para automatizar el funcionamiento de los robots y que éstos puedan interrelacionar con el entorno. Por otro lado, para lograr el control de un robot es necesario aplicar conocimientos de mecánica durante el diseño de la estructura; así como de electricidad, electrónica y sensórica para dar movimiento al robot y conseguir que se adapte y comunique con la informacion del entorno.

Todos estos aspectos hacen de la robótica una materia multidisciplinar que contribuye a la formación tecnológica del alumno, y que requiere de una preparación específica y continua del profesorado de Tecnología en los centros educativos.

2. La robótica en la Legislación Educativa de España

La LOMCE (Ley Orgánica 8/2013, de 9 de diciembre), y las órdenes Educativas posteriores que la han desarrollado, tanto en el ámbito nacional como regional, establecen dos ciclos para la Educación Secundaria Obligatoria (ESO). El primero de ellos comprendería los tres primeros cursos, y el segundo ciclo correspondería en exclusividad a 4o ESO. é ste, a su vez, tendría dos opciones (Enseñanzas Académicas y Enseñanzas Aplicadas). Además, se establecen asignaturas troncales, específicas y de libre configuración autonómica. El currículo de cada una de ellas se desglosa en contenidos, criterios de evaluación y estándares de aprendizaje evaluables. Se trata de adquirir un conjunto de competencias clave.

El Informe PISA (Programme for International Student Assessment, en español Programa Internacional para la Evaluación de Estudiantes), se basa en el análisis del rendimiento de estudiantes a partir de unos exámenes que se realizan cada tres años en varios países, con el fin de determinar sus competencias. En

España, una de las comunidades autónomas con mejores resultados en este Informe es Castilla y León. Por tanto, se muestran las asignaturas que tienen relevancia, en la etapa de Educación Secundaria Obligatoria, para fortalecer el ámbito tecnológico, sin considerar materias básicas como Matemáticas o Física.

2.1. Asignaturas Tecnológicas en Castilla y León

Las asignaturas de ámbito tecnológico (Tecnología, Tecnología de la Informacion y la Comunicación (TIC), Educación Plástica, Visual y Audiovisual (EPVyA) y robótica), en el currículo de Castilla y León, quedan distribuidas de la siguiente manera: (Figura 1)

Materias	PRIMER CICLO (1º, 2º y 3º ESO)			SEGUNDO CICLO (4º ESO)	
	Períodos lectivos semanales				
	1º	2º	3º	Enseñanzas Académicas	Enseñanzas Aplicadas
TRONCALES DE OPCIÓN					
Tecnología					4
ESPECÍFICAS					
Tecnología	3		3		
Tecnologías de la Información y la Comunicación				2	2
LIBRE CONFIGURACIÓN AUTONÓMICA					
Tecnología				2	
Control y Robótica			2		
Programación Informática				2	2

Figura 1: Asignaturas de ámbito tecnológico en Castilla y León con la LOMCE.

2.2. La Asignatura Control y robótica en Castilla y León

En la Comunidad de Castilla y Leo n no hay unos contenidos mínimos específicos de robótica en la asignatura de Tecnología, si bien, el profesorado puede establecerlos en su correspondiente Programación Didáctica. No obstante, en la citada ORDEN EDU/589/2016, de 22 de junio [5], se establece el currículo de una importante materia sobre la Robotica en la ESO que se denomina Control y robótica.

A través de esta asignatura se integran conocimientos relacionados con las matematicas, ciencias experimentales y tecnologías de la informacion y la comunicación, los cuales toman una mayor significación al ser orientados hacia la resolución de un problema tecnológico.

Los contenidos de la asignatura Control y robótica, materia de libre configuracion autonomica en Castilla y León para el curso académico 2016/17, se agrupan en cuatro bloques:

Sistemas automáticos de control

- Sistemas automáticos de control. Definición y componentes característicos: captadores, comparadores, controladores y actuadores.

- Representacion gráfica de sistemas automaticos de control.
- Necesidades y aplicaciones de los sistemas automáticos de control. Ámbito industrial y domótica.

robótica

- Origen y evolución de la robótica. Clasificación general de los robots. Aplicaciones de los robots.
- Arquitectura de un robot: sensores, actuadores, microprocesador y memoria.
- Tipos de sensores. Características técnicas y funcionamiento. Circuitos típicos para sensores.
  - Sensores digitales: pulsador, interruptor, de equilibrio.
  - Sensores analógicos: de intensidad de luz, de temperatura, de rotación, optoacopladores, de distancia.
- Actuadores. Características técnicas y funcionamiento. Circuitos típicos para actuadores. Tipos de actuadores: zumbadores, relés, motores de corriente continua, servomotores, LEDs, pantallas LCD.
- Movimientos y localización: grados de libertad (articulaciones), sistemas de posicionamiento para robot.
- Características de la unidad de control compatible con software libre. Conexión de sensores y actuadores con la unidad de control. Tipos de entradas y salidas (analogicas y digitales).
- Configuración del proceso de impresión: control, calibración y puesta a punto de impresoras 3D.
- Comunicación con el ordenador: tipos de conexión alámbrica e inalámbrica (wifi, bluetooth y telefonía móvil).

Programación y Control

- Software libre de control a través de programación visual con bloques. Diagramas de flujo: simbología. Bloques de programacion. Estructura secuencial y de control (condicionales y bucles).
- Software libre de control a través de lenguaje textual de programación por codigo: variables, funciones, bucles, operadores aritméticos y compuestos. Lenguajes de alto nivel.
- Software libre y firmware de impresión 3D.
- Gestión de archivos de impresión: descarga de modelos STL. Gestión de archivos gCode.

Proyectos de robótica

- Análisis y definición del problema: necesidades estructurales, mecánicas, electrónicas y energéticas de un robot.
- Diseño del sistema robótico: definición de los parámetros geométricos y dinámicos. Elección de servoaccionamientos. Elección de dispositivos electrónicos y de control.

- Depuración de programas de control. Defectos de precisión: mecanismos de autocorreccion. Proceso de subida del programa de software libre al sistema de control.
- Documentacion técnica de un proyecto. Tipos de licencias para compartir documentación y programas.
- Tipos de impresoras 3D. Técnicas de fabricación. Tipos de materiales empleados.

3. Metodología en robótica Educativa

La robótica puede ser objetivo de conocimiento, recurso pedagogico y también refuerzo tecnológico. Los aspectos metodológicos mas importantes a tener en cuenta son:

1. La referencia metodológica debe ser el proceso de resolución técnica de proyectos, donde se tengan en cuenta las fases de investigación: planteamiento de un reto, análisis de diferentes alternativas de solución, elección de una de ellas, experimentación, evaluación, revisión del resultado final y documentación del proyecto técnico.
2. El grupo-clase deberá organizarse mediante el trabajo en equipo, fomentándose así el aprendizaje cooperativo.
3. Imprescindible contar con software y hardware adecuados, y seguir unas normas relativas al cuidado del material; respeto entre compañeros, utilización adecuada de los recursos, etcétera.

3.1. Distribución de Espacios

En la sala de informática (Figura 2) se requieren una serie de adaptaciones sencillas, para que la metodología sea eficiente. No obstante, en la práctica nos podemos encontrar con diversos impedimentos (presupuesto limitado, espacio reducido o falta de previsión). Interesa disponer de proyector en el techo y pantalla en la zona de la pizarra. Una mesa central para realizar con éxito el trabajo colaborativo y diseñar, montar y probar los robots. Se pueden acercar las sillas si se precisa. En esta mesa debe haber un ordenador que facilite las explicaciones del profesor sobre las aplicaciones y programas informáticos, a través de la proyección para todos los alumnos. éstos orientarán sus sillas cuando corresponda. En un armario rodante, con cerradura, se guardarán los robots, la impresora 3D o cualquier dispositivo.

3.2. Recursos Materiales y Técnicos

Es importante una adecuada metodología, pero imprescindible contar con medios que permitan desarrollarla con éxito. En el aula deberíamos poder contar con algunos de estos recursos (Figura 3):

- Simuladores informáticos de robótica educativa.

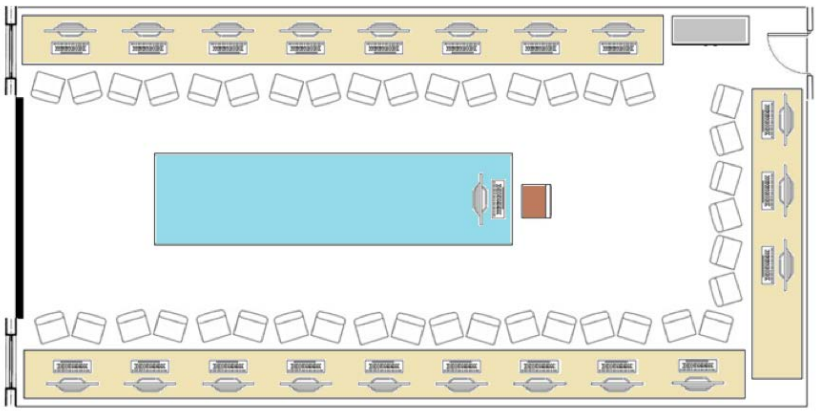


Figura 2: Distribución de espacios en el aula de informática y robótica para realizar trabajo individual y colaborativo.

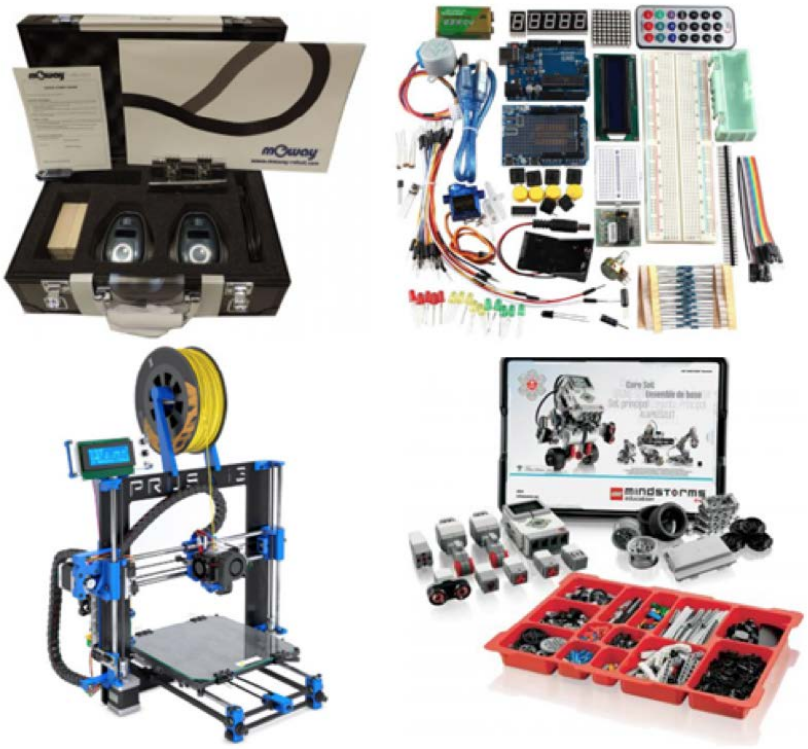


Figura 3: Diferentes dispositivos empleados en robótica (robots mOway, sensores, impresora 3D, kit de Lego).

- Programa informático de diseño 3D.
- Aplicaciones informáticas para facilitar la programación en diferentes lenguajes (FMSLOGO, Processing, Scratch, S4A, ArduBlock, Visualino...).
- Programas informáticos para la simulación de circuitos eléctricos.
- Programas informáticos para animación 2D y 3D.
- Controladoras.
- Equipos informáticos adecuados.
- Cañón y pantalla, para las explicaciones generales del profesor desde el ordenador.
- Impresora 3D.
- Robots educativos (montados o por piezas para su montaje).
- Tarjetas Arduino.
- Placa de pruebas (protoboard), para facilitar las conexiones rápidas y abaratar costes.
- Componentes eléctricos, sensores, actuadores y cables de conexión.
- Kits robóticos, preparados para la docencia.

4. Evolución de la robótica

Desde antiguo se han construido maquinas imitando las partes del cuerpo humano. Los antiguos egipcios ya unían brazos mecánicos a sus estatuas, para hacer creer que su movimiento era inspiración divina, cuando en la realidad eran manipulados por sacerdotes. Los griegos emplearon sistemas hidráulicos para el movimiento de algunas estatuas de sus templos, para conseguir la fascinación de los adoradores.

4.1. Clasificación de los Robots por Generaciones

La clasificacion mas común es la que agrupa los robots por generaciones. Se establecen 4 generaciones a día de hoy. Se empieza a hablar de la Quinta Generación, considerando que tendrá nanotecnología con inteligencia artificial global, utilizará modelos de conducta y una nueva arquitectura de subsunción (estructura en capas donde cada una de ellas presenta una capacidad independiente, dotando al robot de una funcionalidad simple). Tendrán mucho que proponer las nuevas generaciones de estudiantes interesados en esta materia (Figura 4).

4.2. Clasificación de Robots según su Arquitectura

Se consideran los robots poliarticulados, móviles, androides, zoomórficos e híbridos. Estos últimos (Figura 5) se corresponden con aquellos de difícil clasificación, cuya estructura es combinación de algunas de las anteriores. Por ejemplo, robot humanoide y con ruedas; o una estructura formada por un carro móvil con brazo robótico articulado, empleado para desactivación de explosivos por los TEDAX.

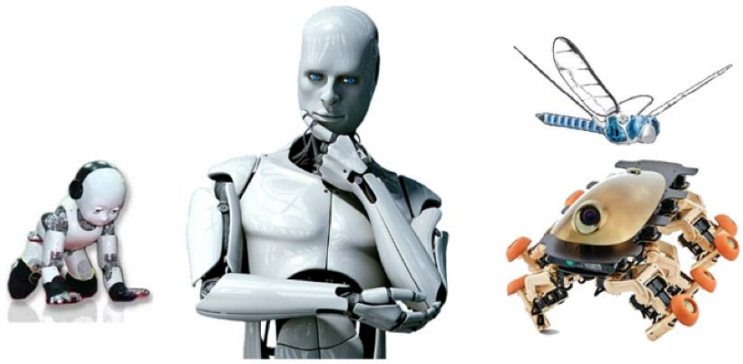


Figura 4: Quinta Generación de robots. Se incorporarán modelos de conducta.

Figura 5: Robots híbridos.

4.3. Otras Clasificaciones

Pueden establecerse nuevas clasificaciones de los robots, siendo las más empleadas[2]:

- Clasificación de los robots según su sistema de control.
- Clasificación de los robots según su nivel de control.
- Clasificación de los robots según el nivel de su programación.
- Clasificación de los robots según su nivel de inteligencia.
- Clasificación de los robots según su finalidad.

4.4. Partes de un Robot

Las partes constituyentes de un robot son las siguientes (Figura 6):

- Estructura mecánica.
- Sistemas de control (circuito eléctrico, microcontrolador, radiocontrol, control inalámbrico directo).
- Fuente de energía.
- Sensores.
- Actuadores.
- Programación.

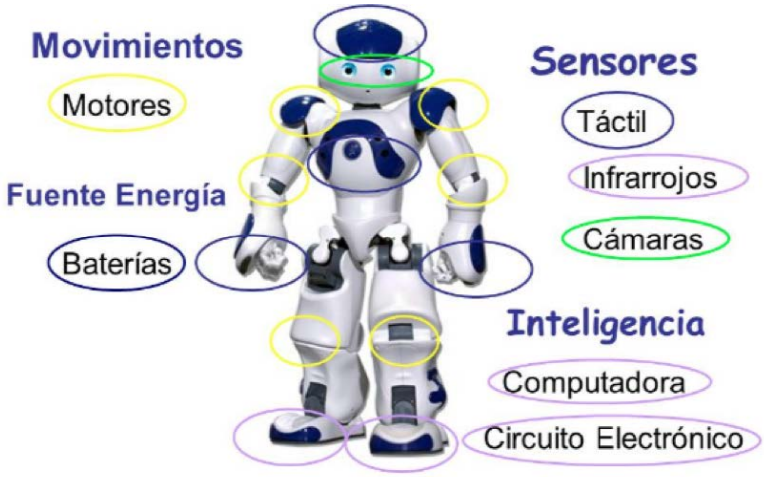


Figura 6: Partes de un robot.

5. Sensores

Un sensor es un dispositivo que proporciona información a la computadora de lo que ocurre en el entorno o en el robot que está siendo controlado. Convierte magnitudes físicas en parámetros cuantificables. Los sensores permiten conocer la localización del robot, su ambiente de trabajo, sus condiciones de trabajo, los objetos con los que debe interactuar, sus propios parámetros físicos... Los sensores que más se emplean en robótica son los siguientes:

- Sensor de contacto.
- Sensor de infrarrojos.
- Sensor de temperatura.
- Sensor de luz.
- Sensor de ultrasonidos.

A continuación se representan las características del sensor de ultrasonidos y su conexión a una placa Arduino (Figura 7):

6. Actuadores

Un actuador es un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado. é ste recibe la orden de un regulador o controlador, después genera la orden para activar un elemento final de control como, por ejemplo, una válvula. Existen varios tipos de actuadores (electrónicos, hidráulicos, neumáticos). Se describen sus características más destacadas en la memoria. Entre los más empleados en robótica destacan los motores y servomotores.

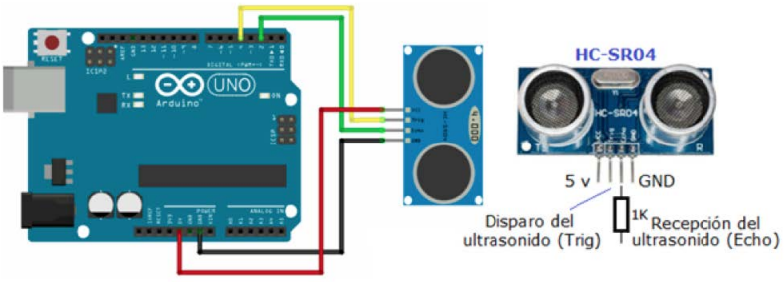


Figura 7: Sensor de ultrasonidos y ejemplo de conexión a la placa Arduino UNO.

7. Controladoras

Las controladoras son el cerebro del robot. Se encargan de reconocer la información que viene del exterior a través de los sensores y hacer funcionar los actuadores conforme al programa almacenado en su memoria. Las controladoras son dispositivos de entrada/salida, para facilitar la interaccion con el entorno. Se componen de entrada, salida y conexión al ordenador. En el trabajo se describen controladoras sin memoria (BSP, ENCONOR...) y con memoria. Entre éstas, tiene mucha implantación en robótica la placa Arduino.

8. Kits Robóticos Comerciales

Se describen productos robóticos educativos, especialmente configurados para el aprendizaje en esta materia como el robot mOway, PrintBot Evolution (empresa BQ) o los kits WeDo y MindStorms de LEGO.

9. Lenguajes de Programación

Los lenguajes de programación tienen importancia clave en el proceso de enseñanza-aprendizaje de la robótica en el aula. Nos permiten la comunicación entre el usuario y el ordenador. En el campo robótico, la programación será la información que permita que las diferentes partes del robot estén comunicadas y coordinadas. Los lenguajes mas empleados en la Robotica Educativa son: programación logo, programación Processing, programación Scratch y programación ARDUINO y S4A. Se presentan sus características y ejemplos prácticos resueltos de cada uno. Si pretendemos una comprensión global del conjunto de actividades que requiere la robótica, debemos cubrir las siguientes etapas con nuestros alumnos (Figura 8):

- Realización de un programa con Scratch/S4A, que resuelva un reto robótico planteado.

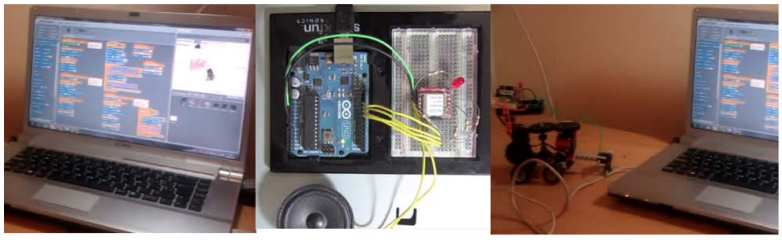


Figura 8: Etapas básicas de la pedagogía robótica (programación, prueba sobre placa y conexión al robot).

- Prueba de nuestro programa, con placa Arduino (u otras similares), conectada por USB al ordenador. Habremos conexionado sensores/actuadores a la placa ayudándonos de otra placa de pruebas (protoboard).
- Montaje del circuito sobre un robot, controlándole desde el ordenador con USB, antes de realizar las pruebas con pila externa. En este caso el robot sería autónomo.

10. Diseño e Impresión 3D

Aquí describo aplicaciones informáticas empleadas en diseño 3D y fabricación digital, adecuadas para la enseñanza. Es importante que los docentes vayan incorporando estos recursos tan innovadores como pedagógicos. Los programas descritos, relativos al diseño 3D fueron Sketchup, Inkscape, TinkerCAD, Autodesk 123D Design, OpenSCAD, FreeCAD y TopMod.

Por otra parte, se explica la aplicación informática CURA, empleada en impresión 3D. Es interesante considerar que los pasos que deben seguirse son: diseño, exportar el archivo a formatos compatibles con la impresora 3D (STL, gCode), imprimirlos y montarlos. Este proceso se aprecia en la imagen (Figura 9).

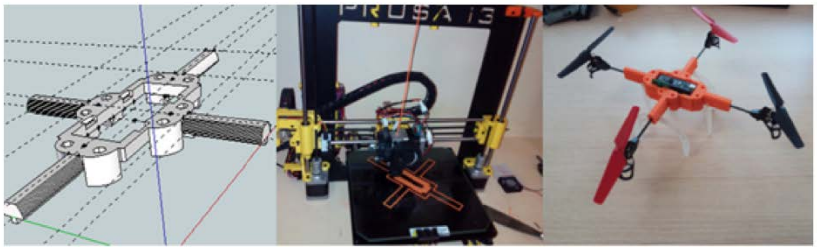


Figura 9: Procesos de diseño de una pieza de robot, impresión 3D y montaje final.

Se presenta el diseño de piezas 3D de un dron, para su posterior impresión y montaje. También se incluyeron todos los dispositivos electrónicos necesarios. El aspecto final fue el siguiente (Figura 10):

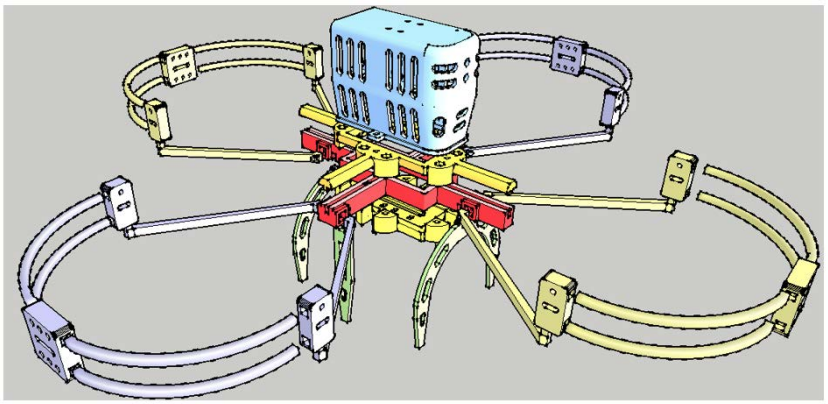


Figura 10: Vista de conjunto del montaje final 3D, de todas las piezas mecánicas del dron.

11. Programas de Simulación

Para el aprendizaje de las disciplinas técnicas comienza con la explicación de sus fundamentos. Posteriormente, conviene emplear programas que permitan simular procesos y experiencias. Hace años no se disponía de estas herramientas, y sólo era posible reproducirlas físicamente en el laboratorio. Hoy día, hay a nuestra disposición aplicaciones que podemos emplear.

En el Trabajo Fin de Máster se describen los siguientes programas:

- Simulación en electricidad y electrónica: Crocodile Clips, Fritzing y Autodesk 123D Circuits.
- Simulación robótica: BrazoRobot y Webots.
- Programas de animación: Pivot Animator (2D), Synfig Studio (2D) y Blender (3D).

12. Redes Neuronales Profundas en robótica

Cuando las redes neuronales profundas (Deep Neural Network, DNN) se comenzaron a utilizar en reconocimiento de patrones, sus primeras capas ocultas no avanzaban lo suficiente en su aprendizaje sobre la información de entrada. En muchos casos no estaban aprendiendo nada en absoluto. Debido a la naturaleza del algoritmo de entrenamiento de las redes neuronales, muchas veces los datos

de partida se alojaban cerca de su inicialización aleatoria. El uso de diferentes técnicas, ha permitido que estas redes profundas sean hoy más potentes (Figura 11).

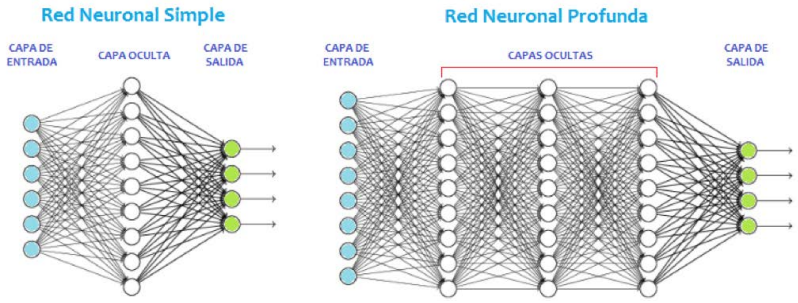


Figura 11: Distribución de capas en una Red Neuronal Simple y en una Red Neuronal Profunda.

La imagen (Figura 12) muestra que una red neuronal profunda es capaz de aprender. En el nivel más bajo, la red se fija en los patrones relevantes de contraste local. La siguiente capa los procesa, fijándose en los rasgos que se asemejan a las referencias. Por último, la capa superior aplica esos rasgos faciales para hacer plantillas. Una red neural profunda es capaz de componer sucesivas características complejas en cada una de sus capas posteriores. Se induce un aprendizaje automatizado sobre un conjunto de datos y características. Esta aplicación de redes neuronales profundas se ha enfrentado con éxito a modelos sobre representaciones de imágenes, audio, escritura y también sobre actividad molecular[1].

También las redes neuronales aportan un enfoque prometedor para reducir la fragilidad del robot y favorecer su adaptación al entorno. En un artículo publicado en Nature[7], han presentado el primer robot capaz de adaptarse a las averías. El estudio titulado Robots that can adapt like animals (Robots capaces de adaptarse igual que los animales), enseñan cómo construir un robot capaz de recuperarse automáticamente de la pérdida de alguna o varias de sus seis patas en menos de dos minutos. Su algoritmo de aprendizaje (Prueba y Error Inteligente) permite la adaptación a los daños sin necesidad de planes de contingencia autodiagnóstico o pre-especificados.

13. Realidad Aumentada en robótica Educativa

La realidad aumentada (RA) es un área creciente en la investigación de realidad virtual. El entorno del mundo que nos rodea proporciona una gran cantidad de información que es difícil duplicar en un ordenador. Un sistema de realidad aumentada genera una vista compuesta para el usuario. Es una combinación

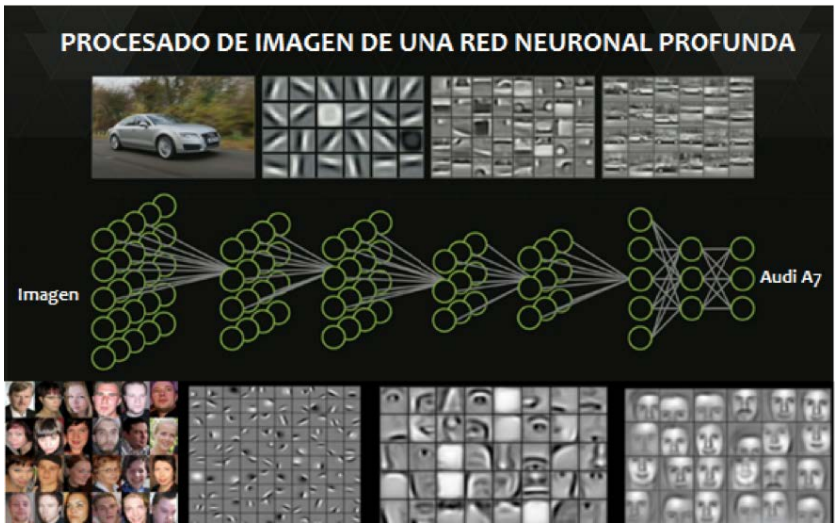


Figura 12: Visualización de la evolución interpretativa de imágenes en cada capa oculta de una DNN.

entre la escena real y una escena virtual generada por el ordenador que aumenta la percepción con información adicional[4].

Un equipo de investigadores de la Agencia de Ciencia y Tecnología de Japón, ha desarrollado un modelo de control remoto para robots que podría ser uno de los más avanzados hasta el momento. Se trata de TouchMe[3], una interfaz de realidad aumentada para controlar un robot y hacerle realizar tareas a distancia mientras se lo observa en tercera persona. Como si se tratara de un videojuego, tocas la pantalla y el programa te muestra el movimiento que realizará el robot antes de que lo ejecute, evitando cometer errores. Por último, se muestra una imagen referida al empleo de realidad aumentada (RA) para el manejo de un robot rastreador de competición[6]. Sobre la imagen que observamos en la tableta (Figura 13), correspondiente al circuito del suelo, se proyectan unos puntos de referencia rojos, asociados a tres balizas para la triangulación de la posición del robot. Así podremos determinar su velocidad y aceleración.

La barra superior contiene 12 puntos luminosos relativos a diferentes sensores. Estos puntos pueden estar blancos (10 de ellos en la imagen) o negros (2 de los 12). La diferencia se debe al estado en que se encuentre cada sensor. También se suministra el nivel de batería, la velocidad (m/s) y la aceleración (m/s2).

14. Conclusiones y Líneas de Trabajo Futuras

Con este trabajo, se pretendió transmitir una visión amplia, tanto de todas las disciplinas tecnológicas que intervienen en la configuración de la robótica para el proceso de enseñanza-aprendizaje del alumnado, como para la investigación y

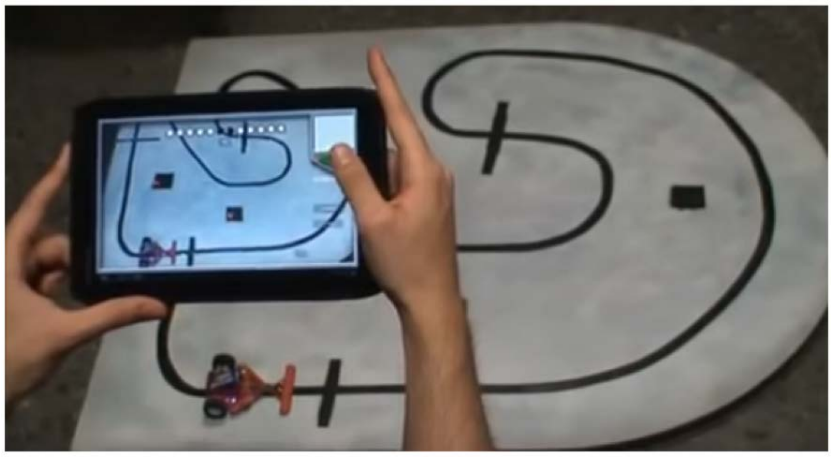


Figura 13: Manejo de un robot de competición, empleando realidad aumentada para su control.

desarrollo de nuevas técnicas. La robótica es tanto un objetivo de conocimiento, un recurso pedagógico como un refuerzo tecnológico.

Todos estos aspectos hacen de la robótica una materia multidisciplinar que contribuye a la formación tecnológica del alumno, y que requiere de una preparación específica y continuada del profesorado de Tecnología en los centros educativos.Además, se destaca la importancia que las redes neuronales profundas y la realidad aumentada están teniendo en la robótica. Son áreas de investigación que aportan avances, y lo seguirán haciendo en el futuro, para que la interacción con el entorno y la operatividad del usuario sean cada vez más eficientes.

Llegados hasta aquí, quedan nuevos y apasionantes retos, y líneas futuras de investigación todavía por descubrir. De momento, interesaría contar con prácticas bien diseñadas sobre programación y puesta a punto de robots, procedimientos sencillos para la instalación y manejo del software Arduino y S4A, creación de un lenguaje de programación de alto nivel compatible y de un hardware estándar para diferentes usos educativos, proyecto de realidad aumentada para control de robots, DNN para mejorar su interacción...

Referencias

1. Aprendizaje profundo. `https://www.datarobot.com/blog/a-primer-on-deep-learning/` Último acceso: Julio-2016.
2. Evolución de la robótica. `http://www-pagines.fib.upc.es/~rob/protegit/treballs/Q2_03-04/general/kind.htm` Último acceso: Julio-2016.
3. Interfaz ra touchme. `http://www.neoteo.com/touchme-controlando-robots-con-realidad-aumentada` Último acceso: Julio-2016.
4. Introducción a la realidad aumentada. `http://www.se.rit.edu/~jrv/research/ar/introduction.html` Último acceso: Julio-2016.

5. Legislación educativa de la comunidad de castilla y león:. http://www.educa.jcyl.es/dpburgos/es/informacion-especifica-dp-burgos/inspeccion-educativa/normativa-lomce, Último acceso: Julio-2016.
6. Realidad aumentada para controlar un robot. https://youtu.be/Z43aaSiPgHg Último acceso: Julio-2016.
7. Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

Sistemas conexionistas: Aplicación a la predicción del mercado bursátil

Pablo Vicente Juan[1] y Angélica González Arrieta[1]

Departamento de Informática y Automática, Universidad de Salamanca. Plaza de la Merced s/n. 37008, Salamanca, España
{pablovicente93,angelica}@usal.es

Resumen La predicción de la tendencia de los mercados bursátiles ha despertado gran interés tanto en investigadores como inversores debido a la compleja naturaleza del problema y a las posibilidades de obtener altos retornos. Dicha predicción puede servir como recomendación a corto plazo así como como aviso inicial de la evolución del mercado a largo plazo. Por ello, la línea de trabajo seguida se centra en la investigación de diferentes técnicas conexionistas y de machine learning para la predicción de las variaciones en el IBEX 35. Se comprobará la eficiencia del modelo propuesto mediante la comparación del retorno obtenido por dicho modelo frente a otros algoritmos que no utilizan técnicas inteligentes.

Keywords: Machine learning, trading, correlación, tendencia bursátil, predicción, IBEX 35.

1. Introducción

El análisis de la tendencia de los mercados es uno de los factores más importantes a la hora de desarrollar estrategias de inversión. Dichas estrategias se traducen en carteras de valores que consisten en una combinación determinada de activos financieros en los cuales se invierte. La predicción de tendencias permite crear carteras balanceadas minimizando el riesgo y maximizando el retorno.

La importancia de los mercados sigue en alza como muestra el aumento del volumen monetario intercambiado. El índice DJIA ha multiplicado por 8 su volumen de operaciones hasta llegar a alcanzar los 7.5 $ billones mientras que el mercado de divisas ha llegado a alcanzar los 5 $ trillones en un solo día [1]. Estos hechos ponen de manifiesto la necesidad de su estudio minucioso.

En la actualidad los inversores han tenido más en cuenta el riesgo a la hora de tomar decisiones debido a la brusquedad de los cambios acontecidos en los mercados bursátiles en tiempos recientes. Existen diversas teorías e hipótesis que ponen de manifiesto la dificultad adicional de conseguir retornos como la hipótesis de los mercados eficientes y del *Random Walk* [2]. Sin embargo, la inteligencia artificial ha rebatido dichas teorías convirtiéndose en herramienta vital en cualquier firma que pretenda mantener beneficios de tiempos pasados.

La evolución acontecida en este campo ha permitido mejorar los resultados debido a tres razones principales. El desarrollo de potentes algoritmos conexionistas y otras técnicas de machine learning que permiten realizar cálculos más

precisos, la evolución acontecida en el hardware que ofrecen entrenar modelos de mayor complejidad en un tiempo menor y la disponibilidad de los datos tanto históricos como en tiempo real.

Debido a la evolución explicada, este trabajo abordará el estudio de diferentes técnicas conexionistas y de machine learning para el proceso de predicción de las variaciones en el IBEX 35. Se procesarán diferentes indicadores económicos como índices bursátiles, commodities o divisas para después desarrollar un modelo de predicción de la tendencia del IBEX 35 utilizando diferentes clasificadores. Se diseccionará el conjunto de datos en diferentes periodos y se utilizarán distintas combinaciones de variables en los experimentos realizados. Se diseñará un modelo de trading sencillo y se comparará el retorno obtenido con otras estrategias que no aplican técnicas predictivas.

El trabajo se divide de la siguiente manera: en la sección 2 se hace una revisión del estado del arte, en la sección 3 se presenta el modelo y los experimentos desarrollados, en la sección 4 se presentan los resultados, en la sección 5 el modelo de trading y en la 6 las conclusiones y líneas futuras de trabajo.

2. Revisión del estado del arte

Los primeros modelos implementados se basaban en técnicas conexionistas como la redes neuronales, que debido al desarrollo del algoritmo de retropropagación y las redes neuronales profundas han permitido mejorar tanto la forma en que se entrenan como su capacidad de abstracción.

Shaun and Ruey [3] investigan la posibilidad de entrenar una red neuronal que tuviese como entrada 500 índices con valores de los últimos 20 años. El sistema es capaz de predecir y ajustarse a sí mismo realizando la comparación entre el valor predicho y el valor real de los índices. Por contra, las redes neuronales producen ciertos valores con un error muy alto que no son abordados.

Naeini *et al* [4] comparan el rendimiento de perceptrones multicapa (MLP) y redes neuronales recurrentes de Elman en el proceso de predicción de stocks utilizando datos históricos. Llegan a la conclusión de que las redes neuronales MLP son mucho mejores en la predicción de la cantidad de cambio mientras que las redes de Elman predicen la dirección del cambio con mayor acierto.

Saad *et al* [5] realizan un estudio comparativo de diferentes variantes de las redes neuronales como *time delay*, recurrentes y probabilísticas para predecir la tendencia del Dow Jones. Concluyen que los tres métodos obtienen resultados comparables a corto plazo, si bien dichos resultados son poco prometedores.

A pesar de algunos resultados obtenidos remarcables, las propuestas anteriores presentan problemas de *overfitting*, haciendo que la eficiencia de las redes decaiga rápidamente en muestras no exploradas. La evolución del machine learning ha permitido solucionar este problema mejorando los resultados.

En [6], Shen *et al* utilizan una máquina de soporte vectorial (SVM) para calcular la tendencia de los principales índices americanos. Para ello utilizan las correlaciones entre los distintos índices bursátiles como valores de entrada. Una vez calculada la tendencia, modelan un sistema que es capaz de conseguir beneficios superiores a dos modelos matemáticos simples en el mercado americano sin

especificar los periodos y tipos de entrenamiento utilizados. Huang *et al* evalúan la capacidad de las SVMs para predecir la evolución del índice NIKKEI 225 y la comparan con redes neuronales [7]. Después de realizar el estudio, proponer la combinación de SVM con otros métodos de clasificación por encima del resto. Sin embargo, los resultados no superan el 60 % de acierto.

Moody and Saffel [8] utilizan un algoritmo de *Recurrent Reinforcement Learning* (RRL) para la obtención y optimización de estrategias de inversión. Este tipo de sistemas abstrae las características del mercado y únicamente modela una serie de acciones que tienen una recompensa. El algoritmo tiene que alcanzar un cierto objetivo utilizando las acciones disponibles en función del conocimiento que tenga del medio. Por tanto, cuanto mayor sea el entrenamiento mayor será la posibilidad de obtener beneficios. Debido a que el algoritmo aprende a base de errores y grandes cantidades de información, puede provocar pérdidas inesperadas ante situaciones no presentadas.

3. Modelo propuesto para la predicción bursátil

El modelo tratará el problema como una clasificación en el que hay que predecir las subidas (1) o bajadas (0) del IBEX 35 entre dos días determinados.

3.1. Extracción de los datos

Los datos han sido obtenidos de la página quandl.com siendo el rango temporal desde el 07/07/1993 hasta el 20/04/2016 [9]. La tabla 1 muestra los distintos valores que serán analizados para su inclusión en el modelo de predicción. Todos los valores que no correspondan a índices se toman en $. Las divisas cotizan en un valor continuo perteneciendo el valor recogido a primera hora de la mañana.

Tipo de datos	**Nombre**
Stock index	ASX, Hang Seng Index, Nikkei 225, DAX, CAC 40, IBEX 35, Dow Jones Industrial Average, SP 500
Commodity	Oro, Plata, Platino, Barril de Brent
Currency	AUD-USD, USD-GBP, JPY-USD

Tabla 1: Variables objeto de estudio.

3.2. Alineamiento de secuencias

El país de pertenencia de cada uno de los mercados determina el horario de actividad de los mismos. Los usos horarios no coinciden debido a la franja horaria de cada uno de los mercados, como muestra la tabla 2, por lo que es necesario un proceso de alineamiento de las series temporales utilizadas.

Bolsas de valores			Zona horaria		Horario local		UTC	
Tipo	**País**	**Ciudad**	**Zona**	**Δ**	**Apertura**	**Clausura**	**Apertura**	**Clausura**
Índice ASX	Australia	Sidney	AEST	+10	10:00	16:00	0:00	6:00
Índice Nikkei 225	Japón	Tokio	JST	+9	9:00	15:00	0:00	6:00
Índice HSI	Hong Kong	Hong Kong	HKT	+8	9:00	15:00	0:00	6:00
Índice DAX	Alemania	Frankfurt	CET	+1	8:00	20:00	7:00	19:00
Índice CAC 40	Francia	Paris	CET	+1	9:00	17:30	8:00	16:30
Índice IBEX 35	España	Madrid	CET	+1	9:00	17:30	8:00	16:30
Currency	Reino Unido	Londres	GMT	0	7:00	15:00	7:00	15:00
Commodity (LBMA)	Reino Unido	Londres	GMT	0	10:30	15:00	10:30	15:00
Commodity (LPPM)	Reino Unido	Londres	GMT	0	9:45	15:00	9:45	15:00
NYSE	Estados Unidos	Nueva York	EST	-5	9:30	16:00	14:30	21:00
NASDAQ	Estados Unidos	Nueva York	EST	-5	9:30	16:00	14:30	21:00

Tabla 2: Franjas horarias de los mercados.

El proceso empleado consiste en alinear las fechas del resto de series respecto al DJIA de manera que las fechas de las series sean iguales. El proceso implica la creación de valores en días en los que ciertos mercados no abren, así como la eliminación de valores para los que el DJIA no cotiza. Para cubrir los huecos que aparecen al alinear las secuencias se aplica la interpolación lineal, que obtiene mejores resultados que la eliminación de valores [10].

3.3. Clasificadores

La predicción de este tipo de series temporales es una tarea compleja debido a la gran cantidad de variables que influyen en el resultado. Por ello, se ha decidido hacer un estudio de las técnicas más conocidas ya que cada una de ellas aporta unas ventajas diferentes al resto. Entre dichas técnicas se encuentran: Redes Neuronales (ANN), Logistic Regression (LRC), Linear and Quadrant Discriminant Analysis (LDA y QDA), Bernoulli Naive Bayes (NBB), K-Nearest Neighbors (KNN), Support Vector Machines (SVM), AdaBoost (ABC), Bagging (BGC) y Randon Forest (RFC).

Los modelos propuestos han necesitado adecuar los parámetros al problema tratado. Sin duda, las redes neuronales han necesitado de mayor énfasis en el proceso de modelado de su arquitectura. El número de entrada de la red viene determinada por el número de características empleado, mientras que el número de neuronas de salida será 1, variando entre el valor 0 (bajada) o 1 (subida). Para lograr la salida en el rango [0,1] se ha elegido la función sigmoide en la capa de salida, mientras que se ha utilizado la tangente hiperbólica en la capa oculta. Respecto a la capa oculta se han estudiado arquitecturas con distinto número de neuronas obteniéndose resultados similares. Finalmente, se ha decidido utilizar 5 neuronas si hay menos de 10 variables de entrada y 50 en caso contrario.

Respecto al resto de modelos, sus parámetros han sido elegidos utilizando una técnica conocida como *grid search*. Una vez se han establecido los posibles valores para los parámetros de cada modelo, esta técnica prueba todas las posibles combinaciones de los mismos sobre el conjunto de datos con el fin de elegir aquellos que ofrecen mejor resultado. El hecho de realizar esta búsqueda

responde a la influencia que tiene en la predicción cada ajuste realizado y que puede provocar desajustes no deseados en los algoritmos. Por tanto, la búsqueda exhaustiva garantiza la elección de los parámetros correctos acordes tanto con la teoría como con las características propias de los datos.

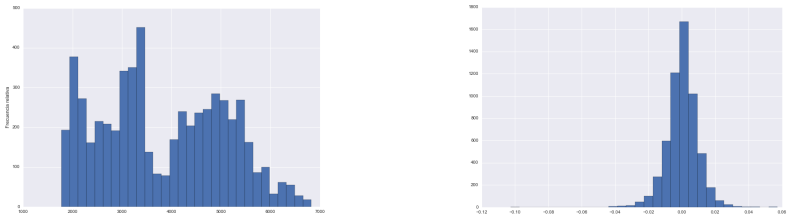
3.4. Selección de características

Cada una de las series temporales tiene un rango de valores muy dispar que dificulta el proceso de predicción del modelo. Por lo que es necesario normalizar el rango y la distribución de valores y eliminar su tendencia utilizando los siguientes métodos.

Log return Mide las ganancias ocurridas durante un periodo de tiempo como logaritmos de los valores del inicio y fin del periodo (fig. 1).

$$r_i = log(\frac{p_t}{p_{t-k}}) \tag{1}$$

donde p_t indica el precio en el día t y p_{t-k} en el día $t-k$. El valor de k indica la diferncia de días entre los valores.



(a) Variable original (b) Retorno logarítmico

Figura 1: Distribución de frecuencias para el valor de cierre del índice AXJO.

Diferencia relativa El segundo de los formatos utilizados para el modelado de la variación del valor consiste en calcular la diferencia relativa entre valores.

$$RD_t = \frac{p_t - p_{t-k}}{p_{t-k}} \tag{2}$$

$$RDP_t = \frac{p_t - r_{t-k}}{p_{t-k}} * 100 \tag{3}$$

donde RD_t es la diferencia de precios y RDP_t es la diferencia en porcentaje.

Escalado a la unidad La última forma de modelar la diferencia consiste en la utilización de únicamente dos valores. El 1 para codificar los incrementos entre dos periodos de tiempo y el -1 o 0 para codificar los decrementos. Para realizar dicho cálculo los valores serán escalados a la unidad conservando su signo.

$$RDB_t = \frac{p_t - p_{t-k}}{\|p_t - p_{t-k}\|} \tag{4}$$

Como se ha comentado, los mercados se encuentran interconectados haciendo que los movimientos de uno afecten a otros. Por ello, será necesario el estudio de la correlación entre las distintas variables respecto del IBEX (fig. 2). La correlación mide la similitud entre la evolución de dos variables a lo largo del tiempo estando su resultado en el intervalo [-1, 1].

$$(f * g)[l] = \sum_{t=-\infty}^{\infty} f * [t]g[t-k] \tag{5}$$

donde $f(t)$ es una serie temporal, $g(t)$ es la segunda serie temporal, t es el tiempo y k es el desplazamiento.

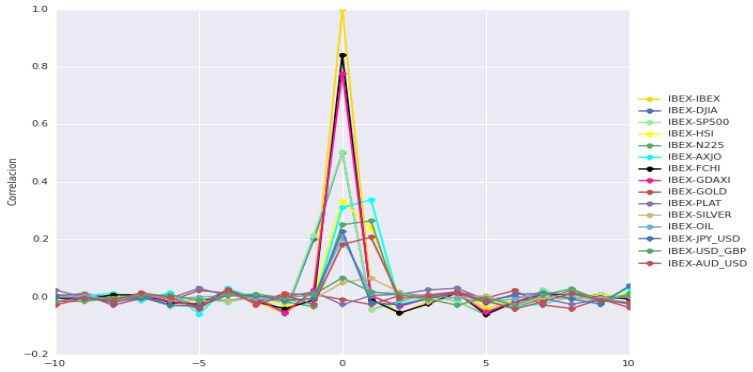


Figura 2: Correlación cruzada de las variables estudiadas respecto del IBEX 35.

La correlación del IBEX 35 alcanza el valor máximo para el día actual, sin embargo, los valores previos no se encuentran correlacionados. El índice francés (FCHI) y el alemán (GDAXI) tienen una alta correlación para el valor actual. Por contra, su valor en el día actual no podrá ser usado ya que no está disponible ya que abre a la misma hora que el IBEX. Los índices DJIA y SP500 muestran una correlación moderada para valores del día anterior que sí pueden ser utilizados. Por su parte, los índices HSI, N225 y AXJO muestran una ligera correlación para el mismo día y debido a su localización horaria sí que se pueden usar dichos valores para emitir la predicción del mismo día. Valores como el oro o el petróleo muestran correlaciones moderadas para el mismo día pero no están disponibles en el momento necesario.

Aunque la correlación de las variaciones diarias ha demostrado ser moderada, es necesario considerar la correlación entre variaciones a largo plazo. Si en lugar de la diferencia entre los días t y t-k con k=*1*, se varía el valor de k entre 1 y 30 días se obtiene la correlación en el largo plazo. Esta variación de k aumenta la distancia entre los días observándose relaciones en intervalos de tiempo mayor.

La figura 3 muestra la correlación de varios índices bursátiles estudiados para distintos valores de k. Se observa que cuanto mayor es el valor de k, mayor es la correlación. Según [6], este comportamiento es achacable a que el incremento del valor k produce que las salidas se solapen incrementando la correlación, eliminando el ruido y haciendo dicha relación entre variables más clara.

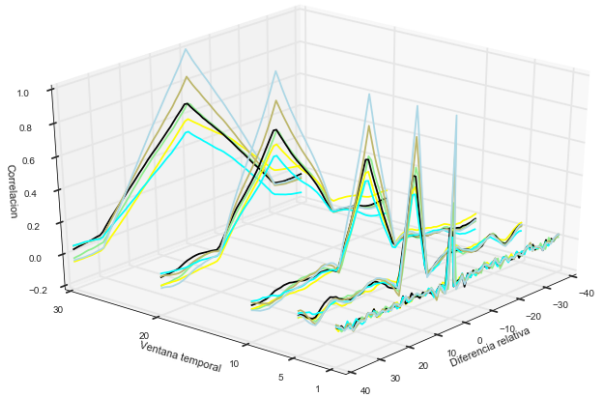


Figura 3: Correlación cruzada para distintos valores de k.

3.5. Experimentos

Predicción del día siguiente El objetivo del modelo consiste en predecir la tendencia del valor del día siguiente utilizando el valor del día anterior. Siendo $x_i(t)$ el valor de la serie i, donde $i \in \{1, 2, ..., N\}$, en el instante de tiempo t y $N = 14$, cada una de las diferencias entre días se define como:

$$X_i(t) = f(x_1(t), x_1(t-k)) \tag{6}$$

siendo k igual a 1, ya que el objetivo consiste en predicciones entre días consecutivos, y f la función que calcula el retorno logarítmico o la diferencia relativa. El nuevo vector de características V_t de cada instancia se define como:

$$V_t = (X_1(t), X_2(t), ..., X_N(t)) \tag{7}$$

Siendo Y el valor de cierre diario del IBEX 35 a predecir, la variable objetivo se define como:

$$Y(t) = \begin{cases} 1 & si \quad \frac{IBEX_t - IBEX_{t-k}}{\|IBEX_t - IBEX_{t-k}\|} = +1 \\ & \qquad\qquad\qquad\qquad siendo\ k = 1 \\ 0 & si \quad \frac{IBEX_t - IBEX_{t-k}}{\|IBEX_t - IBEX_{t-k}\|} = -1 \end{cases} \tag{8}$$

El objetivo del modelo consiste en predecir Y_{t-1} utilizando el vector de características V_t. Para realizar la predicción el modelo clasifica la tendencia entre $t+1$ y t utilizando la tendencia entre t y $t-1$. Este desplazamiento de $t=-1$ de la variable objetivo es necesario ya que no se dispone de los valores de todos los índices x_i en el instante t para predecir el valor de Y en t.

La estrategia experimental, recogida en la tabla 3,consiste en realizar pruebas utilizando distintas variables en función de su significado económico.

Tests	**Características**
Test 1	1 Log return de índices con k=1
	2 Log return de índices con k=1,2,3,5,10,30
	3 Diferencia relativa de índices con k=1
	4 Diferencia relativa de índices con k=1,2,3,5,10,30
	5 Diferencia relativa en porcentaje de índices con k=1
	6 Diferencia relativa en porcentaje de índices con k=1,2,3,5,10,30
	8 Diferencia relativa binaria de índices con k=1
	8 Diferencia relativa binaria de índices con k=1,2,3,5,10,30
Test 2	1 Log return de commodities con k=1
	2 Log return de commodities con k=1,2,3,5,10,30
	3 Diferencia relativa de commodities con k=1
	4 Diferencia relativa de commodities con k=1,2,3,5,10,30
	5 Diferencia relativa en porcentaje de commodities con k=1
	6 Diferencia relativa en porcentaje de commodities con k=1,2,3,5,10,30
	7 Diferencia relativa binaria de commodities con k=1
	8 Diferencia relativa binaria de commodities con k=1,2,3,5,10,30
Test 3	1 Log return de todas con k=1
	2 Log return de todas con k=1,2,3,5,10,30
	3 Diferencia relativa de todas con k=1
	4 Diferencia relativa de todas con k=1,2,3,5,10,30
	5 Diferencia relativa en porcentaje de todas con k=1
	6 Diferencia relativa en porcentaje de todas con k=1,2,3,5,10,30
	7 Diferencia relativa binaria de todas con k=1
	8 Diferencia relativa binaria de todas con k=1,2,3,5,10,30

Tabla 3: Conjunto de test con distintas características seleccionadas.

Hasta ahora los experimentos diseñados se basan en la agrupación de características en función de su naturaleza económica. Sin embargo, en el último conjunto de experimentos se utilizan otras técnicas en el proceso de selección.

La profundidad de ciertas características dentro de un árbol de decisión puede ser utilizada para valorar la importancia relativa de dicha característica respecto a la variable objetivo. Las características situadas en la parte superior del árbol tienen mayor influencia en el proceso de predicción y son utilizadas en otro de los experimentos. Por otra parte, se utilizará *Principal Component Analysis* (PCA) como técnica de reducción de la dimensionalidad. La transformación se realiza de tal manera que el primer componente acumula la mayor cantidad de varianza posible, mientras que cada uno de los siguientes añaden la mayor cantidad de varianza posible no considerada en el componente anterior.

Predicción a largo plazo La predicción del valor del día siguiente permite realizar inversiones en el corto plazo, sin embargo, se ha comprobado que la correlación aumenta cuando la diferencia relativa entre días lo hace. El hecho de la eliminación del ruido y la existencia de tendencias más definidas cuando la diferencia temporal entre dos días es mayor posibilita obtener predicciones a largo plazo más acertadas. La formulación matemática responde a la enunciada en la predicción del día siguiente modificando el valor de k en función del lapso de tiempo empleado. Si anteriormente se utilizaba el valor de k igual a 1, ahora se realizarán pruebas para 3, 5, 10, 20 y 30. Por tanto, en este conjunto de experimentos se va a predecir la tendencia entre el valor de mañana respecto al valor de varios días atrás.

Periodos de prueba Los datos recogidos suman un total 5740 muestras para todo el periodo. Aunque se realizarán pruebas dividiendo el conjunto en un 70 % de muestras para training y 30 % para testing, la solución más utilizada cuando se trata de series temporales se conoce como *sliding window*. Esta técnica consiste en segmentos de instancias consecutivas utilizadas para training y testing de manera alternativa. El uso de esta técnica se debe a dos motivos: los valores pasados del mercado solo tienen influencia en el futuro inmediato por lo que no tiene sentido utilizar periodos de entrenamientos con datos excesivamente pasados. Por otro lado, la ventana de testing no debería ser superior al 20 % de la de training ya que el régimen del mercado cambia rápidamente.

En función de los dos métodos de entrenamiento presentados se diseñan pruebas utilizando el 70 % de las muestras para entrenamiento y el 30 % para pruebas. Por otro, lado se realizan pruebas utilizando el método de la ventana deslizante con un periodo de entrenamiento de 7 años y dos periodos de pruebas distintos de 1 año y 1 mes.

Métricas de evaluación Una correcta evaluación de los resultados es crítica a la hora de evaluar el modelo ya que tiene que ser capaz de medir el error de manera adecuada. La medida utilizada para validar el sistema se conoce como *accuracy* y que mide la cantidad de elementos correctamente clasificados sobre el

total. Es posible utilizar esta medida debido a que contamos con un conjunto de datos balanceado, es decir, el número de muestra de cada clase está equilibrado.

4. Resultados de los modelos de predicción

Cada una de las variantes del modelo explicada en la sección anterior ha sido entrenada y validada siguiendo los esquemas propuestos. Los resultados permite comprobar la existencia de ciertas estrategias capaces de emitir predicciones con un alto porcentaje de acierto.

Predicción a largo plazo La tabla 4 muestra el resultado del conjunto de experimentos realizados utilizando el método clásico. Se observa que ninguno de los algoritmos empleados es capaz de conseguir resultados superiores al resto ni superar el 60 % de acierto. Por otra parte, ninguno de los conjuntos de características empleados en el proceso muestra ser superior. Teniendo en cuenta que las posibles clases a predecir son únicamente dos, simplemente lanzando una moneda podríamos conseguir un 50 % de probabilidades de éxito haciendo que el resultado obtenido no aporte ningún beneficio.

Test	ANN	LRC	LDA	QDA	SVM	NBB	KNN	ABC	BGC	RFC
1.1	53,90 %	56,07 %	57,08 %	57,45 %	53,73 %	54,77 %	53,85 %	56,11 %	53,41 %	56,33 %
1.2	55,18 %	56,44 %	56,65 %	52,92 %	53,43 %	53,43 %	54,83 %	56,11 %	53,86 %	56,95 %
1.3	52,28 %	56,59 %	56,80 %	54,17 %	54,33 %	54,77 %	52,49 %	54,87 %	52,06 %	56,62 %
1.4	55,44 %	55,92 %	56,13 %	51,07 %	53,05 %	53,43 %	51,38 %	54,87 %	53,22 %	56,78 %
1.5	53,13 %	57,10 %	56,98 %	57,49 %	53,92 %	54,77 %	53,36 %	56,11 %	53,43 %	56,75 %
1.7	51,36 %	57,40 %	57,33 %	52,86 %	53,69 %	53,43 %	52,05 %	56,11 %	54,86 %	58,21 %
1.8	49,77 %	55,10 %	55,10 %	55,10 %	54,77 %	54,77 %	54,36 %	55,38 %	54,82 %	54,91 %
2.1	52,81 %	54,18 %	54,18 %	54,59 %	52,66 %	52,04 %	50,36 %	54,13 %	47,67 %	50,74 %
2.2	53,81 %	53,71 %	54,32 %	50,33 %	50,92 %	51,06 %	52,52 %	54,13 %	50,03 %	52,43 %
2.3	56,54 %	53,55 %	53,67 %	52,84 %	50,68 %	51,84 %	52,19 %	52,34 %	51,04 %	50,75 %
2.4	51,43 %	53,79 %	53,77 %	50,71 %	52,10 %	50,56 %	52,43 %	52,32 %	48,68 %	52,25 %
2.5	53,35 %	54,29 %	54,29 %	52,13 %	51,96 %	51,84 %	50,35 %	54,13 %	50,77 %	50,63 %
2.6	52,91 %	53,38 %	53,38 %	50,69 %	51,25 %	51,06 %	52,12 %	54,13 %	50,03 %	52,66 %
2.7	49,12 %	51,99 %	51,99 %	52,27 %	51,84 %	51,84 %	48,55 %	54,13 %	52,63 %	51,99 %
2.8	54,18 %	54,43 %	54,48 %	51,12 %	51,03 %	51,06 %	51,76 %	54,13 %	52,10 %	54,46 %
3.1	56,51 %	55,82 %	54,82 %	52,24 %	52,65 %	53,97 %	53,66 %	56,11 %	53,11 %	54,45 %
3.2	55,23 %	56,90 %	55,19 %	51,13 %	53,63 %	51,92 %	50,64 %	56,11 %	53,82 %	56,62 %
3.3	52,12 %	52,15 %	55,38 %	49,25 %	47,61 %	53,77 %	51,98 %	54,87 %	51,64 %	54,34 %
3.4	53,37 %	56,00 %	55,89 %	54,34 %	55,31 %	52,70 %	53,50 %	56,11 %	54,77 %	57,14 %
3.5	51,00 %	56,01 %	55,77 %	47,53 %	47,80 %	53,77 %	49,44 %	56,11 %	53,02 %	54,68 %
3.6	53,45 %	56,00 %	55,89 %	54,34 %	55,31 %	52,70 %	53,50 %	56,11 %	54,77 %	57,14 %
3.7	55,29 %	54,83 %	54,87 %	52,86 %	53,62 %	53,77 %	50,69 %	55,38 %	51,18 %	55,40 %
3.8	56,93 %	55,68 %	55,82 %	53,71 %	52,68 %	52,70 %	52,67 %	55,38 %	53,12 %	56,23 %

Tabla 4: Conjunto de pruebas para entrenamiento 70 % y 30 %.

El mismo conjunto de pruebas se ha realizado para la estrategia de la ventana deslizante. En concreto, se ha establecido un periodo de entrenamiento para un total de 7 años y dos periodos de pruebas de un año y de un mes. El proceso se ha repetido moviendo la ventana de manera que los resultados obtenidos consisten en la media del total de iteraciones. Los resultados obtenidos para los dos casos de pruebas no mejoran al entrenamiento clásico anterior. Este hecho permite afirmar que el método de entrenamiento de la ventana deslizante no obtiene mejores resultados, si bien, tampoco peores. Al utilizar este tipo método es posible agilizar el entrenamiento ya que no es necesaria tanta información en el entrenamiento. Por otra parte, es posible afirmar que periodos de pruebas más cortos no mejoran los resultados en este caso.

El hecho de contar con multitud de características ha permitido diseñar un conjunto de pruebas amplio. Sin embargo, los tests realizados hasta ahora han ofrecido resultados poco prometedores ya que tantas características pueden llegar a confundir al clasificador. Por ello, se ha decidido utilizar técnicas de selección de características y reducción de dimensionalidad cuyos resultados para el mejor algoritmo en cada caso se muestran en la tabla 5. Se observa que estas técnicas no consiguen mejorar los resultados obtenidos en los experimentos anteriores lo que confirma la dificultad de predecir la tendencia entre días consecutivos, ni siquiera haciendo un tratamiento de las variables.

Tipo de selección	**Accuracy**
Árboles de decisión. Entrenamiento clásico	51,12 %
Árboles de decisión. Entrenamiento 2 años, testing 1 año	59,25 %
PCA 3 componentes. Entrenamiento 2 años, testing 1 año	47,08 %
PCA 50 componentes. Entrenamiento 2 años, testing 1 años	58,37 %
PCA 100 componentes. Entrenamiento 2 años, testing 1 año	60,12 %

Tabla 5: Resultados mediante selección de características y PCA.

Predicción a largo plazo Los resultados de predecir la diferencia entre el valor de mañana y el de un cierto número de días atrás se muestran en este apartado. En este caso, la tarea consiste en predecir la diferencia en binario entre el día $t+1$ y el día $t-k$ donde k es el número de días anteriores al actual. Haciendo uso de la notación empleada a la hora de describir las variables el problema consiste en predecir $RDB_{(t+1)-(t-k)}$ utilizando la variables $RDB_{(t)-(t-k-1)}$.

La gráfica 4 muestra la evolución en la tasa de acierto de las predicciones emitidas en función de la diferencia de días utilizando el método sliding window. Se observa una clara tendencia creciente que mejora las predicciones. Cuando se establece k mayor o igual a 20 días la precisión comienza a estabilizarse. Se observa que Bernoulli Naive Bayes obtiene la mejor precisión con un total de 86 % de acierto, mientras que SVM y LDA consiguen una mayor estabilidad a lo largo del tiempo.

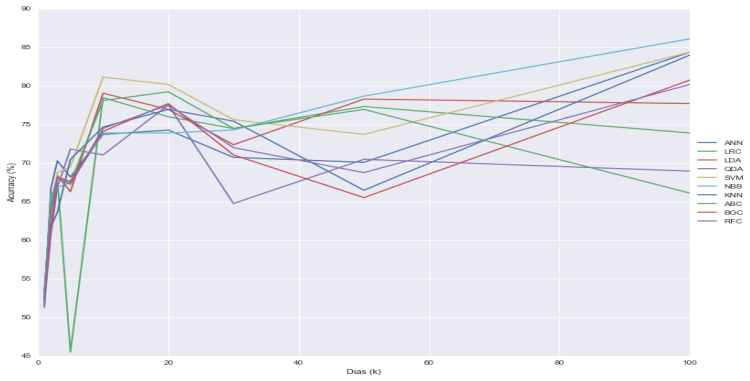


Figura 4: Accuracy de los distintos modelos en función del valor k.

En este caso las variables utilizadas influyen de manera importante en la predicción obtenida. Los valores óptimos se obtienen cuando únicamente se utiliza la diferencia relativa binaria como variable de entrada para el mismo k que se utiliza para calcular la variable objetivo con desplazamiento $t = -1$. Ni siquiera es necesario los valores de las divisas o de las commodities para alcanzar este valor, hecho que agiliza el entrenamiento. Por otra parte, si que es cierto que no es necesario utilizar periodos de entrenamiento extensos.

5. Modelo de trading automático

Con el fin de analizar la posibilidad de obtener rendimientos a partir de dichas predicciones, se ha desarrollado un algoritmo que realizan operaciones de compra y venta en función del índice. Si suponemos un producto financiero cuyo valor varía en función de los movimientos del IBEX 35, sería interesante saber los momentos óptimos para comprar y vender dicho producto. Para ello, se compara la rentabilidad de los tres modelos en cuatro periodos.

Cada modelo partirá con un capital inicial de 10000 $ que se invertirá en función de la estrategia seguida. Se establecerán 4 periodos de tiempo distintos de un total de 60 días cada uno en los que se utiliza las diferentes estrategias. Al final del periodo se deberá vender todas las acciones y se cuantificará el beneficio.

Modelo 1 Este modelo hace uso de una estrategia *buy and hold* clásica. El dinero inicial será invertido en comprar el producto, que varía en función del IBEX, el primer día del periodo para venderlo el último día. El resultado dependerá de la tendencia del IBEX para ese periodo según la siguiente ecuación:

$$retorno = C_i * \frac{V_t - V_o}{V_o} \tag{9}$$

donde C_i es el capital inicial, V_t es el precio final de la acción y V_o el inicial.

Modelo 2 El segundo modelo realiza las operaciones de compra y venta de dicho producto en función de la diferencia de precios entre días contiguos. Si el precio del índice de hoy es superior al de ayer, se asume que mañana será superior a hoy y se invierte el capital disponible en el producto, en caso contrario, se vende.

Modelo 3 Utiliza las predicciones realizadas por el clasificador para reaccionar a la variación del índice. Si la predicción del día siguiente es alcista (1), se invierte el capital en dicho producto, en caso de predicción bajista (0) se vende . Hay que tener en cuenta que la predicción de la tendencia del día de mañana se realiza en función de un valor de varios días atrás y verificar que el valor de hoy no es inferior al de k días atrás.

La tabla 6 muestra los porcentajes de beneficios en cada periodo así como el acumulado por cada modelo. Aunque el modelo inteligente únicamente es superior en 2 de los 4 periodos, su capacidad para reducir el riesgo en los periodos donde otros sufren grandes pérdidas le permiten obtener un resultado final muy superior al resto. Los periodos con mayor variación del precio provocan mayor inestabilidad en el precio ya que no manejan con precisión dichos cambios.

Periodos	Modelo 1	Modelo 2	Modelo 3
26-01-2004 - 21-04-2004	3.12 %	-0.18 %	8.09 %
17-04-2008 - 14-07-2008	-15.67 %	-11.94 %	1.00 %
18-04-2011 - 14-07-2011	-7.21 %	2.52 %	-4.08 %
25-01-2016 - 20-04-2016	6.76 %	-4.86 %	6.19 %
Acumulado	-3.25 %	-14.46 %	11.2 %

Tabla 6: Porcentaje de retornos obtenidos.

A pesar de haber conseguido retornos superiores al resto de modelos, es necesario tener en cuenta que no se han aplicado las tasas correspondientes. El hecho de aplicarse una tasa sobre los beneficios implica un descenso en mayor medida de los ingresos del modelo propuesto debido a que sus ganancias son superiores.

6. Conclusiones y líneas de trabajo futuras

La investigación llevada a cabo ha permitido demostrar la posibilidad de predecir la tendencia de IBEX 35 en función de distintas variables económicas.

Se han elegido varios de los valores que más peso tienen en la economía realizándose un procesamiento de los mismos alineándolos y realizando la interpolación en los huecos. Se han creado variables modelando el retorno y la diferencia relativa para un análisis de las tendencias.

El estudio de los distintos algoritmos ha servido para crear diversos modelos y comprobar la eficacia de cada uno de ellos. Las divisiones temporales han

permitido comprobar que una mayor cantidad de instancias en el conjunto de entrenamiento no mejoran los resultados.

Se ha comprobado que las predicciones a corto plazo está sujetas al ruido inherente del mercado provocando variaciones difíciles de predecir ya que no responden a movimientos deterministas. Sin embargo, la predicción a largo plazo ha obtenido una alta tasa de acierto.

El diseño de diversas estrategias de inversión ha facilitado la comparación entre el modelo inteligente diseñado respecto de otros clásicos, demostrando que los modelos inteligentes mejoran los rendimientos obtenidos.

Aunque se han conseguido realizar una predicción de la tendencia y se ha diseñado un modelo de trading exitoso, existen ciertos aspectos que necesitan ser investigados. La utilización de las variables descritas permite una predicción con un gran nivel de acierto a largo plazo. Si bien, sería conveniente el estudio de variables macroeconómicas tanto propias como externas.

El valor de los diferentes índices bursátiles responde a la capitalización de las empresas más relevante dentro de una bolsa o sector de la misma. Por ello, la inclusión de la evolución del valor de cotización de dichas empresas en el modelo podría mejorar los valores de predicción del IBEX 35.

La mejora del modelo de trading automático recae sobre la inclusión de posiciones en corto en el mismo, es decir, la posibilidad de vender valores para después comprarlos.

Sin lugar a dudas, la aplicación de inteligencia artificial en el campo financiero tiene un exitoso porvenir. No solo permiten adelantarse a los movimientos del mercado sino que pueden ser integradas en modelos automáticos.

Referencias

1. Monetary and E. Department, "Foreign exchange turnover in april 2013: preliminary global results," tech. rep., Bank for International Settlements, 2013.
2. B. G. Malkiel and E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.
3. S.-I. Wu and R.-P. Lu, "Combining artificial neural networks and statistics for stock-market forecasting," in *Proceedings of the 1993 ACM conference on Computer science*, pp. 257–264, ACM, 1993.
4. M. P. Naeini, H. Taremian, and H. B. Hashemi, "Stock market value prediction using neural networks," in *Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on*, pp. 132–136, IEEE, 2010.
5. E. W. Saad, D. V. Prokhorov, D. C. Wunsch, *et al.*, "Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks," *Neural Networks, IEEE Transactions on*, vol. 9, no. 6, pp. 1456–1470, 1998.
6. S. Shen, H. Jiang, and Z. T., "Stock market forecasting using machine learning algorithms." 2012.
7. W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Computers & Operations Research*, vol. 32, no. 10, pp. 2513–2522, 2005.
8. J. Moody and M. Saffell, "Learning to trade via direct reinforcement," *Neural Networks, IEEE Transactions on 12.4*, vol. 12, no. 4, pp. 875–889, 2001.

9. 06 2016.
10. S. Zemke, "On developing a financial prediction system: Pitfalls and possibilities," *Proceedings of DMLL-2002 Workshop at ICML-2002*, 2002.

Autores

María del Pilar Arista Flores, nacida en Lima (Perú), completó la base de su educación superior en el año 2006 obteniendo el título de Graduado en Ingeniería Informática por la Pontificia Universidad Católica del Perú. Profesional con más de 8 años de experiencia en soluciones de Inteligencia de Negocios. En este campo, ha liderado proyectos y realizado consultoría en organizaciones de diversos sectores, principalmente financieras y de consumo masivo. Amplió su formación realizando el Máster Oficial en Sistemas Inteligentes en 2016 en la Universidad de Salamanca. Su trabajo se han centrado en el desarrollo de sistemas de recomendación utilizando técnicas de minería de datos. Actualmente busca complementar sus conocimientos, participando de proyectos de Big Data en diferentes empresas de Perú.

Alejandro Benito Santos natural de Salamanca, es Graduado en Ingeniería Informática por la Universidad de Salamanca. En 2011 fue agraciado con una beca ARGO del Ministerio de Educación y Ciencia que le permitió lanzar su carrera profesional en Londres (Reino Unido), especializándose en el desarrollo de aplicaciones móviles iOS. En su trabajo como consultor móvil ha colaborado con diferentes empresas y startups de Silicon Valley y Reino Unido. Recientemente el Big Data y en espacial la visualización de datos han llamado su atención, disciplinas en las que centra su actividad. Otras áreas de su interés son las tecnologías P2P, las criptomonedas y las Humanidades Digitales.

Francisco Javier Blanco Rodríguez es Licenciado en Ciencias Físicas y Doctor por la Universidad de Salamanca desde el año 2003.En la actualidad es Profesor Contratado Doctor en el Área de Ingeniería de Sistemas y Automática en el Departamento de Informática y Automática de la Universidad de Salamanca. Imparte varias asignaturas de robótica tanto en titulación de Grado como de Master. En cuanto a la investigación está interesado en temas de Robots autónomos, robótica educativa y nuevas tecnologías aplicadas en la educación.

Pâmella Arielle Brito de Aquino, natural de Salvador de Bahía, Brasil, completó su base de educación superior en el año de 2010 obteniendo el título de Graduada en Informática por la Universidade Católica de Salvador. En la Universidad de Salamanca amplió sus formación realizando el Máster Oficial en Sistemas Inteligentes en 2015. Sus trabajos se han centrado fundamentalmente en la minería de opiniones basada en análisis de sentimientos.

Gerardo Andrés Corado Juárez, nacido en Guatemala, completó la base de su educación en el año 2012 obteniendo el titulo de Ingeniero en sistemas e informática por la Universidad Rafael Landivar. En la Universidad de Salamanca amplió sus conocimientos con el Master Oficial en Sistema Inteligentes.[en] el 2016. Su trabajo se ha basado en el análisis histórico de incidencias de vandalismo en Wikipedia por medio de la visualización.

Juan M. Corchado Rodríguez es catedrático de Ciencia de la Computación e Inteligencia Artificial. Actualmente es Vicerrector de Investigación y Transferencia de la Universidad de Salamanca y Director del Grupo en Biotecnología, Sistemas Inteligentes y Tecnología Educativa de esta misma universidad (http://bisite.usal.es). Doctor en Informática por la Universidad de Salamanca, España (1998) y doctor en Inteligencia Artificial por la University of the West of Scotland, Reino Unido (2000). Es Coordinador Académico del Instituto de Arte y Tecnología de la Animación. Anteriormente ha sido Subdirector de la Escuela Superior de Ingeniería Informática en la Universidad de Vigo (España, 1999-00) e investigador en la Universidad de Paisley (1995-98). Colabora, como investigador, con el Laboratorio Oceanográfico de Plymouth (Reino Unido) desde 1993. Ha dirigido más de 70 proyectos de Inteligencia Artificial con financiación tanto internacional, como nacional o autonómica. Ha dirigido 16 tesis doctorales y es co-autor de más de 300 libros, capítulos de libros, artículos en revistas científicas, etc. la mayoría de ellos presentan aspectos tanto prácticos como teóricos relacionados con los Sistemas Híbridos de Inteligencia Artificial, la Biomedicina, la Inteligencia Ambiental, los Sistemas Inalámbricos y la Oceanografía. Es autor de más de una veintena de propiedades intelectualesy patentes.

Belén Curto Diego es Licenciada en Ciencias Físicas en la rama de Electrónica y Doctora en Ciencias Físicas. Actualmente es Profesora Titular de Universidad. Pertenece al Dpto de Informática y Automática de la Universidad de Salamanca. Imparte clase en las titulaciones de Grado en Ingeniería Informática, Grado en Ingeniería Química, Máster en Ingeniería Informática y Máster en Sistemas Inteligentes sobre Robótica, Arquitectura de Computadores, Control de Procesos y Automatización Industrial. Pertenece al Grupo de Investigación "Robótica y Sociedad" de la Universidad de Salamanca donde realizan trabajos sobre Inteligencia Artificial-Robótica y Automatización Industrial con empresas del sector alimentario, formación médica, nuclear, etc. Es autora de más de 30 artículos en revistas científicas especializadas. Ha dirigido números trabajos de Tesis Doctoral, Trabajos Fin de Máster y Grado a lo largo de su intensa labor docente e investigadora

---

Alberto Encinas Elvira, nacido en Salamanca, completó la base de su educación superior en el año 2014, obteniendo el título de Graduado en Ingeniería en Electrónica Industrial y Automática por la Universidad de Valladolid. Posteriormente amplió su formación realizando el Máster Oficial en Sistemas Inteligentes en 2016. Sus trabajos se han centrado fundamentalmente en el desarrollo de sistemas electrónicos embebidos en el ámbito de la robótica, la domótica y la inmótica.

Eduardo Flores González nació en Salamanca, en 1987. Es Ingeniero Técnico en Informática de Sistemas y Graduado de Ingeniería Informática por la Universidad de Salamanca. En la misma, también ha realizado el Master en Sistemas Inteligentes, especializándose en análisis visual y minería de datos. Ha formado parte del grupo de investigación BISITE, también en la Universidad de Salamanca. Actualmente desarrolla su actividad profesional desarrollando aplicaciones multiplataforma y colaborando en proyectos de gestión de conocimiento.

Angélica González Arrieta es doctora en Informática por la Universidad de Salamanca. Cuenta con una amplia experiencia investigadora en el campo de la Computación Neuroborrosa y las Redes Neuronales. Es Profesora Titular del Departamento de Informática y Automática de dicha Universidad. Actualmente compatibiliza su labor docente e investigadora con la dirección de diversas actividad formativas sobre seguridad informática, colaborando activamente con la Academia de la Policía Nacional de Ávila.

Vivian F. López Batista es profesora titular de la Universidad de Salamanca en el área de Ciencias de la Computación e Inteligencia Artificial. Doctorada en Informática por la Universidad de Valladolid en 1996. Miembro del Grupo de Minería de Datos. Ha realizado investigación en diferentes campos como procesamiento del lenguaje natural, redes neuronales y minería de datos. Tiene 80 artículos publicados en revistas de reconocido prestigio, talleres y actas de conferencias, 20 libros y capítulos de libros y 20 informes técnicos, la mayoría de ellos en estos temas. Miembro del comité organizador y científico de varios simposios internacionales. Fue directora del Máster en Sistema Inteligente y del Programa de Doctorado en Informática y Automática de la Universidad de Salamanca desde junio de 2010 hasta octubre de 2012.

Daniel López Sánchez, natural de la ciudad castellana de Ávila, cursó sus estudios de Grado en Ingeniería Informática en la Universidad de Salamanca. Posteriormente obtuvo el título de Máster en Sistemas inteligentes por la Universidad de Salamanca. En la actualidad cursa estudios de doctorado en el Grupo de investigación BISITE. Sus principales intereses de investigación son los algoritmos para el aprendizaje automático de bajo coste computacional y las aplicaciones del deep learning en el campo de la minería web. Además, colabora en diversos proyectos de investigación en el ámbito de la bioinformática y análisis de redes sociales.

Diego Milla de Castro, nacido en Valladolid, se graduó en Economía por la Universidad de Lausana (Suiza) en 2014. Gracias a su pasión por la tecnología y a los conocimientos matemáticos y estadísticos adquiridos previamente, amplió su competencia en informática y programación de forma autodidacta. Tras dos años de experiencia laboral como desarrollador Full-Stack, en 2016, decidió realizar el Máster Oficial en Sistemas Inteligentes por la Universidad de Salamanca. Sus trabajos se han centrado en la aplicación de algoritmos de Inteligencia Artificial a diversas tareas musicales, tales como la clasificación automatizada de canciones, el desarrollo de un crawler para la descarga de archivos MIDI o la composición automatizada dirigida de melodías.

María N. Moreno García, Catedrática de Universidad del Departamento de Informática y Automática de la Universidad de Salamanca, directora del grupo de investigación en Minería de Datos y Coordinadora del Programa de Doctorado de Ingeniería Informática. Sus áreas de investigación de interés se centran en el desarrollo y aplicación de algoritmos de minería de datos en diferentes dominios como minería web y de medios sociales o apoyo a las decisiones en medicina.

Vidal Moreno Rodilla, es Licenciado en Ciencias Físicas en la rama de Electrónica y Doctor en Ciencias en 1996. Actualmente es Profesor Titular de Universidad del área de Ingeniería de Sistemas y Automática de la Universidad de Salamanca. Ha impartido docencia sobre Robótica, Inteligencia Artificial y Control en niveles de Grado, Master y Doctorado tanto en Salamanca como en el extranjero. Miembro fundador del Grupo de Investigación Reconocido "Robótica y Sociedad", es autor de varias decenas de publicaciones en revistas internacionales fruto de la dirección de una centena trabajos de investigación que incluye tesis doctorales, Trabajos de Fin de Master, etc.

María Navarro Cáceres, estudiante de Doctorado en Ingeniería informática por la Universidad de Salamanca, está graduada en Ingeniería Informática y en Canto y Guitarra en el Conservatorio de Salamanca. Su mayor interés es unir sus dos pasiones, la música y la informática, en el campo de la investigación. Actualmente, se encuentra desarrollando su Tésis que está centrada en la composición musical automática a partir de la inteligencia artificial, utilizando para ello mecanismos como los sistemas multiagente o los algoritmos bioinspirados. En el ámbito de la investigación, María ha participado en proyectos de tecnología educativa, eficiencia enérgetica y en proyectos de Smart Cities.

Belén Perez Lancho es Licenciada en Ciencias Físicas y Doctora por la Universidad de Salamanca desde 1995. Realizó su formación posdoctoral en la Universidad Paris VI durante el curso académico 1996-97 y desde 1998 es Profesora del Área de Ingeniería de Sistemas y Automática en la Universidad de Salamanca. Imparte docencia en las titulaciones de Grado en Ingeniería Informática, Grado en Física y en el Máster en Sistemas Inteligentes. Ha participado en más de 20 proyectos de investigación, principalmente en aplicaciones de sistemas de control inteligente y sistemas multiagente, y ha ocupado cargos de gestión como Secretaria Académica y Vicedecana de la facultad de Ciencias.

Jorge Revuelta Herrero, nacido y crecido en Salamanca, completó sus estudios universitarios con el Grado en Ingeniería Informática de la Universidad de Salamanca. Posteriormente realizó el Máster Oficial en Sistemas Inteligentes durante el curso 2015/2016. Actualmente se encuentra estudiando el Doctorado en Informática de la misma Universidad y trabajando en el Grupo de Investigación BISITE del Departamento de Informática y Automática. Sus trabajos se centran en el desarrollo software iOS, plataformas de análisis de datos Big Data procedentes de redes sociales y la Inteligencia Artificial.

Roberto Therón Sánchez cursó sus estudios de Informática en la Universidad de Salamanca (Diplomatura)y la Universidad de la Coruña (Licenciatura). Tras entrar a formar parte del Grupo de Investigación en Robótica de la Universidad de Salamanca, presentó su trabajo de Tesis recibiendo el Premio Extraordinario de Doctorado. Posteriormente ha obtenido los títulos de Licenciado en Comunicación Audiovisual (Universidad de Salamanca) y Licenciado en Humanidades (Universidad de Salamanca). En la misma Universidad de Salamanca continúa realizando su trabajo de investigador, como encargado del grupo VisUsal (dentro del Grupo de Investigación Reconocido GRIAL) que se centra en la combinación de enfoques procedentes de la Informática, Estadística, Diseño Gráfico y Visualización de Información, para obtener una adecuada comprensión de conjuntos de datos complejos. En los últimos años, se ha dedicado al desarrollo de herramientas de visualización avanzada para datos multidimensionales. En el área de Analítica Visual desarrolla productivas colaboraciones con grupos e instituciones de reconocido prestigio internacional, como el Laboratorio de Ciencias del Clima y del Medio Ambiente (París) o el Centro de Analítica Visual Avanzada de la ONU (Suiza). Es autor de más de 70 artículos en revistas y congresos internacionales.

Pablo Vicente Juan, nacido en Salamanca, completó la base de su educación superior en el año 2015 obteniendo el título de Graduado en Ingeniería Informática por la Universidad de Salamanca. En la misma universidad amplió su formación realizando el Máster en Sistemas Inteligentes en 2016. Su investigación se ha centrado en el campo del machine learning, donde ha desarrollado algoritmos para la aplicación práctica de dicho paradigma.

Ángel F. Zazo es profesor titular del departamento de Informática y Automática. Imparte docencia en niveles de grado, máster y doctorado y ha participado en numerosos proyectos de investigación en convocatorias competitivas de carácter nacional y regional, fruto de los cuales han sido un buen número de artículos publicados en revistas nacionales e internacionales.