Consensus-based Approach for Keyword Extraction from Urban Events Collections

Ana Alves[a,b], Bernardete Ribeiro[a]

[a] CISUC, Department of Informatics Engineering, University of Coimbra, Portugal

[b] Polytechnic Institute of Coimbra

ana@dei.uc.pt, bribeiro@dei.uc.pt

KEYWORD

Machine Learning; Ontol-ogies; Ensem-ble Learning; Key-word Extraction; Con-ditional Random Fields (CRF); Urban Events

ABSTRACT

Automatic keyword extraction (AKE) from textual sources took a valua-ble step towards harnessing the problem of efficient scanning of large document collections. Particularly in the context of urban mobility, where the most relevant events in the city are advertised on-line, it be-comes difficult to know exactly what is happening in a place.

In this paper we tackle this problem by extracting a set of keywords from different kinds of textual sources, focusing on the urban events context. We propose an ensemble of automatic keyword extraction systems KEA (Key-phrase Extraction Algorithm) and KUSCO (Knowledge Unsupervised Search for instantiating Concepts on lightweight Ontologies) and Conditional Random Fields (CRF).

Unlike KEA and KUSCO which are well-known tools for automatic keyword extraction, CRF needs further pre-processing. Therefore, a tool for handling AKE from the documents using CRF is developed. The architecture for the AKE ensemble system is designed and efficient inte-gration of component applications is presented in which a consensus between such classifiers is achieved. Finally, we empirically show that our AKE ensemble system significantly succeeds on baseline sources and urban events collections.

1. Introduction

Nowadays the most relevant events in the city are advertised on-line. However, it often becomes difficult to know exactly what is happening in a place: information is spread across too many websites, often not easily understandable. The result of the World Wide Web (WWW) exponential growth is a huge amount of data chaotically organized, which turns out tasks like accessing, searching and keeping information difficult. With so many data drifting in the Web, most of the times neither labeled nor categorized, finding the desired information is generally a waste of time. Automatic extraction and summarization methods play an essential role to tackle this problem. Therefore, the goal of automatic extraction is to apply the power and speed of computation to the problems of access and discoverability, adding value to information organization and retrieval without the significant costs and drawbacks associated with human indexers.

The ensemble-based approaches that combine multiple classifiers constitute a new breed of nonstationary learning (NSL) algorithms. These algorithms tend to be more accurate, more flexible and some-times more efficient than single classifiers (Kuncheva, 2004). In this paper we propose an ensemble of different classifiers for effectively extracting a set of keywords from small textual sources and show that the proposed ensemble application achieves better performance than the individual component systems (up to a certain extent concerning the differences between the individual classifiers' reliability), obtaining better results than those reported in the last Key-phrase Extraction Contest on SemEval 2010 (Kim *et al.*, 2010).

The proposed approach comprises two supervised, KEA and CRF, and one unsupervised, KUSCO, machine learning keyword extraction methodologies.

The empirical tests were carried out in *Hulth's dataset* of scientific journal paper abstracts, in *Krap's dataset* abstracts from Computer Science domain. Furthermore, for further validating our approach, two collections of documents regarding music personalities' descriptions extracted from Wikipedia and descriptions about events in general, like theatre plays and music concerts, retrieved from YourSingapore[1] web-pages were considered.

This paper is organized as follows. In the Section 2, we present the background on the automatic keyword extraction regarding related work and specific applications. We describe the ensemble learning methodology, in particular, we look at both the keyword extraction component classifiers and the ways to combine them. The proposed approach is presented in Section 3. Section 4 deals with the experimental setup and in Section 5 results are presented and discussed showing the validity of our approach. Finally, in Section 6 we make conclusions and point some lines of future work.

2. Related Work

Automatic Keyword Extraction (AKE) looks at the problem of automatically identifying the relevant words within a document which has been investigated for more than half a century (Luhn, 1958). Such keywords may be used to classify a text or may serve as a concise summary for a given document, while often useful entries for building an automatic index for a document collection (Mihalcea *et al.*, 04). Most of the methods for AKE rely on statistical and linguistic knowledge, meanwhile recent works are more focused on machine learning techniques. In this line, a wide range of methods have successfully been proposed for tasks of AKE (Wang *et al.*, 2006) achieving better results than merely use of statistics or linguistic knowledge about documents.

Examples of statistical methods include word frequency (Luhn, 1958), word co-occurrence (Matsuo *et al.*, 04) and the *TF-IDF* (Term Frequency - Inverse Document Frequency) term weighting model (Robertson, 04). Such methods have proved to be insufficient to overcome such problems by their own, thus another line of automatic extraction methods considered the linguistic features of words.

Hulth (Hulth, 2003) examines different methods of incorporating this knowledge into AKE and considers syntactic features such as part-of-speech (PoS) tags to the classifier looking only at noun phrases to be candidate phrases. In turn, (Plas *et al.*, 2004) showed that by using lexical resources (EDR - electronic dictionary and Princeton University's WordNet) in such a task results in slightly higher performances than by just resorting to a purely statistically based method.

[1] http://www.yoursingapore.com/ (last accessed 07/03/2016)

Supervised algorithms found in (Turney, 1999) classified words as positive or negative examples of keywords, first applying the *C4.5* decision tree induction algorithm and later a custom-developed algorithm, *GenEx*. The authors conclude that by incorporating specialized procedural domain knowledge keyphrases could be better generated.

Perhaps one of the main contributions to the field is KEA that was proposed in (Witten, et al,1999), using the TF-IDF score of a phrase as well as fined tuned features to build a Naive Bayes classifier. An improvement over KEA, called KEA++, has been proposed in (Medelyan *et al.*, 2006) and also takes advantage of semantic information on terms and phrases gleaned from a domain-specific thesaurus. Other approach based on KEA, but relying on *bagged decision trees* instead of Naive Bayes for classification was proposed in (Medelyan *et al*, 2009) and was called *Maui*. I (Sarkar *et al.*, 2010) and (Wang *et al.*, 2006) neural networks have been proposed and their results demonstrate that their method outperforms KEA. In (Li *et al.*, 2010) various classification models are compared in the task of extracting meaningful keywords from extremely short texts like those we find today on social networking services on the Web. They used a set of features to train those models (TF-IDF, linguistic information, relative position, the length of social snippet, document frequency and capitalization) and the best results reported used Gradient Boosting Machine (GMB).

Today CRF, is considered the state-of-the-art sequence labeling method. It was proposed by Lafferty *et al*. (Lafferty *et al.*, 2001) back in 2001 and since then many published works explored this technique. Peng and McCallum (Peng *et al.*, 2006) showed that CRF outperforms other methods at the task of extracting structured information, such as the information related to the authors and citations from a collection of research papers. In (Zhang *et al.*, 2008) CRF was successful proposed for the task of keyword extraction from Chinese scientific papers. Recently, in (Feng *et al.*, 2012) AKE based on a combination of CRFs and a specific document structure was also presented, while the authors argue the results improved dramatically over the existing ones.

With respect to the unsupervised approach they usually consist of ranking each of the candidate keywords using multiple features and heuristics and selecting the top rated ones (Wan *et al.*, 2008), (Matsuo *et al.*, 04). In (Mihalcea *et al.*, 2004) TextRank, a graph-based ranking model based on the co-occurrence relation between words was presented, although other works based on graph mining were also published (Grineva *et al.*, 2009), (Ortiz *et al.*, 2010), (Rose *et al.*,. 2010). Recently, focusing on keyword extraction from small textual sources such as event and product descriptions (often holding between 30 and 60 words), a novel unsupervised keyword extraction approach was proposed in (Timonen *et al.*, 2012), called Informativeness-based Keyword Extraction, that uses clustering and three levels of word evaluation (corpus, cluster and document level) to address the challenges of short documents.

3. Keyword Extraction Tools

In this section, the applications that will be used to build the ensemble for keyword extraction, namely, KEA (Key-phrase Extraction Algorithm), KUSKO (Knowledge Unsupervised Search for instantiating Concepts on lightweight Ontologies) and also CRF (Conditional Random Fields) will be analysed with more detail.

3.1. KEA

KEA (Keyphrase Extraction Algorithm) is considered one of the important contributions to the field of keyword extraction (Witten *et al.*,:1999). It comprises basically three main phases, as illustrated in Figure 1, which will be described next.

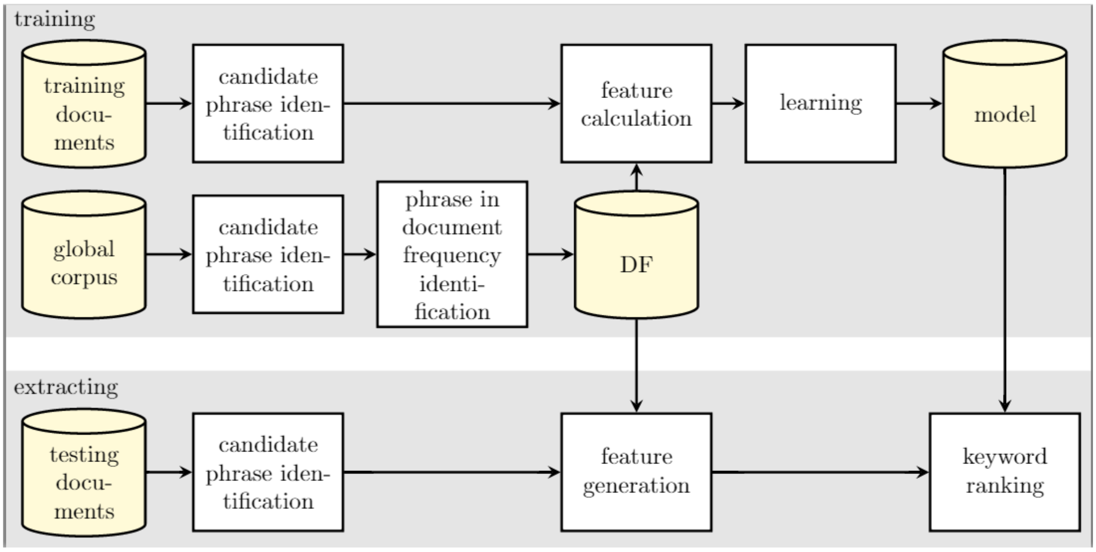
training
training docu- ments
candidate phrase iden- tification
feature calculation
learning
model
global corpus
candidate phrase iden- tification
phrase in document frequency identi- fication
DF
extracting
testing docu- ments
candidate phrase iden- tification
feature generation
keyword ranking

Fig. 1. KEA training and extraction processes (Witten *et al.*, 1999).

First, KEA performs some text preprocessing tasks in order to identify candidate phrases because not all of the phrases in a document are equally likely to be key-phrases *a priori*. The process starts by splitting the input text into words according to the phrase boundaries (e.g. punctuation marks, dashes, brackets, and numbers) where unwanted characters are removed. KEA takes then all the sub-sequences of these initial phrases (by default up to length three), as candidate phrases and eliminates the phrases that begin or end with a *stop-word* or phrases that are a *proper noun*. The stop-word list used by KEA contains 425 words in nine syntactic classes (conjunctions, articles, particles, prepositions, pro-nouns, anomalous verbs, adjectives, and adverbs); finally, these candidate phrases are then case-folded and stemmed.

Second, two specific *attributes* are used to discriminate between key-phrases and not key-phrases: the *TF-IDF* score of a phrase, and the *distance* into the document of the phrase's first appearance (the number of words that precede the first occurrence of the term, divided by the number of words in the document). This corresponds to the *feature calculation* phase.

Finally, in the third phase KEA computes the TF-IDF scores and distance values for all the phrases in the new document, taking the discretization obtained from the training documents. The naive Bayes model is then applied to each phrase and estimates its probability to be a key-phrase. The result is a list of phrases ranked according to their associated probabilities. Assuming that the user may want to extract *r* keyphrases, then KEA outputs the *r* highest ranked phrases (Witten *et al.*, 1999).

3.2. KUSCO

KUSCO (Knowledge Unsupervised Search for instantiating Concepts on lightweight Ontologies) (Alves *et al.*, 2009) is a system that indexes a set of concepts with given Points of Interest (POIs) and Events, semantically enriching them. Generally, KUSCO retrieves information on the Web about POIs and Events and extracts the most relevant terms from these texts. After extraction, those terms are contextualized and enriched with semantic information. Since KUSCO comprises different modules, we will focus on the Meaning Extraction module where *term extraction* from textual descriptions is performed. In Figure2, a schematic visualization of KUSCO's Meaning Extraction module is illustrated.

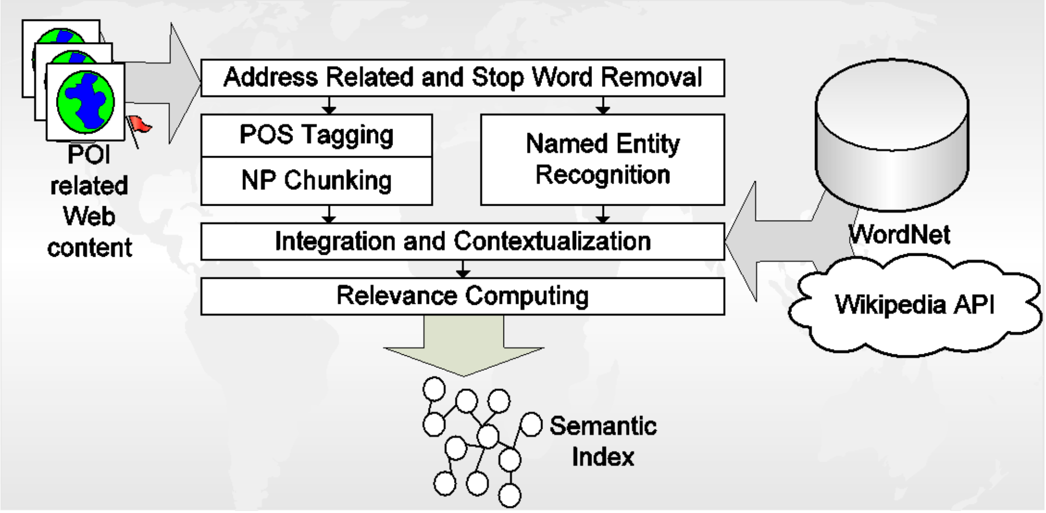

Fig. 2. KUSCO's Meaning Extraction module (Alves *et al.*, 2009).

For each text describing a POI (or an event) the Meaning Extraction module on KUSCO executes a sequence of Natural Language Processing steps to automatically extract the relevant related terms. Each text is broken up into paragraphs, paragraphs into sentences, and sentences into words. Words in a sentence are then tagged by the Brill's Part-of-Speech (POS) tagger (Brill, 1994) which labels each word as a noun, verb, adjective, etc. A Noun Phrase *chunker* (Ramshaw, 1999) is then applied in order to identify every group of words with a head noun which functions together just as a single term. At the same time, the original text is also processed by a Named Entity recognizer (Finkel, 2005) to identify proper names in the text.

As shown in Figure 2, noun phrases (on the left flow, which applies POS tagging and NP chunking) are represented by common nouns while the entities (on the right flow, which applies NER) are represented by proper nouns. Each term in both groups is represented using single or a compound noun, and it is contextualized in lexical resources (WordNet and Wikipedia) which guide the extraction process by validating common-sense terms and which are also used to infer the meaning of each term. These terms are called concepts only after they are contextualized, and their relevance is computed through an extended version of TF-IDF that considers the semantics of each term.

3.3. CRF

CRF (Conditional Random Fields) have been applied in a broad range of areas such as natural language processing, including named-entity recognition (NER), feature induction for NER, identifying protein names in biology abstracts, segmenting addresses in Web pages, information integration word alignment in machine translation, citation extraction from research papers, word segmentation among many other (Sutton *et al.*, 2010). Unlike the majority of methods that do not use most of the features existing in a document, CRF can utilize most of those features sufficiently and effectively for efficient keyword extraction. Experimental results indicate that the CRF model can enhance keyword extraction often outperforming other machine learning methods (Zhang *et al.*, 2008).

In short, CRF is an undirected graphical model that encodes a conditional probability distribution with a given set of features. For the given observation sequential data $X\ (X_1X_2, \dots, X_n)$, and their corresponding status labels $Y\ (Y_1Y_2, \dots, Y_n)$, a linear chain structure CRF defines the conditional probability as follows:

$$P(Y|X) = \frac{1}{Z_x} e^{\sum_i \sum_j \lambda_j f_j (y_{i-1}, y_i, X, i)} \tag{1}$$

where Z_x is a normalization constant, $f_j\ (y_{i-1}, y_i, X, i)$ is a feature function and λ_j is a learned weight associated with feature f_j. The interested reader can find more information in the literature (Lafferty *et al.*, 2001).

4. Ensemble Learning Approaches

As shown in (Escovedo *et al.*, 2014), one of the biggest problems in using a single classifier to address concept drift problems is that when the classifier learns a dataset and then we need it to learn a new one, the classifier must be retrained with all data, or else it will forget everything already learned. Otherwise, using the ensemble, there is no need to retrain it again, because it can keep the previous knowledge and still learn new data. We believe it can be applied to the problem of automatic keyword extraction, since new textual descriptions about events comes up in a daily basis.

Following Kuncheva's work (Kuncheva, 2004) four approaches (A, B, C and D) for ensemble methods are usually considered (see Figure 3). Each one focuses on a different level of action which concerns combining the results of multiple methods in order to get improved results. At the combining level (Approach A) different ways can be used to combine the results from the classifiers. Many ensemble paradigms employ the same classification model, for example, a decision tree or a neural network, but there is no evidence that this strategy is better than using different models (Approach B). At feature level (Approach C) different feature subsets can be used for the classifiers, either if they use the same classification model or not. Finally, the data sets can be modified so that each classifier in the ensemble is trained on its own data set (Approach D).

Hulth (Hulth, 2004) presented an algorithm for AKE combining statistical and linguistic methods, showing that the number of incorrect assigned keywords could be highly reduced, by combining then the predictions of several classifiers. Using an ensemble of Neural Networks, Wang et. al (Wang *et al.*, 2006) depicted a method where the key-phrase extraction is viewed as a crisp binary classification task, training the neural network ensemble to classify whether a phrase is key-phrase or not.

To discriminate between positive and negative examples, the following features (or attributes) of a phrase in a given document are adopted: its *term frequency*, whether they *appear in the title*, *abstract* or *headings* (subheadings), and its *frequency* appearing in the paragraphs of the given document, i.e., the distribution of a phrase in a given document. Later, Zhang (Zhang, 2009) combined successfully several machine learning models to extract keywords from Chinese documents.

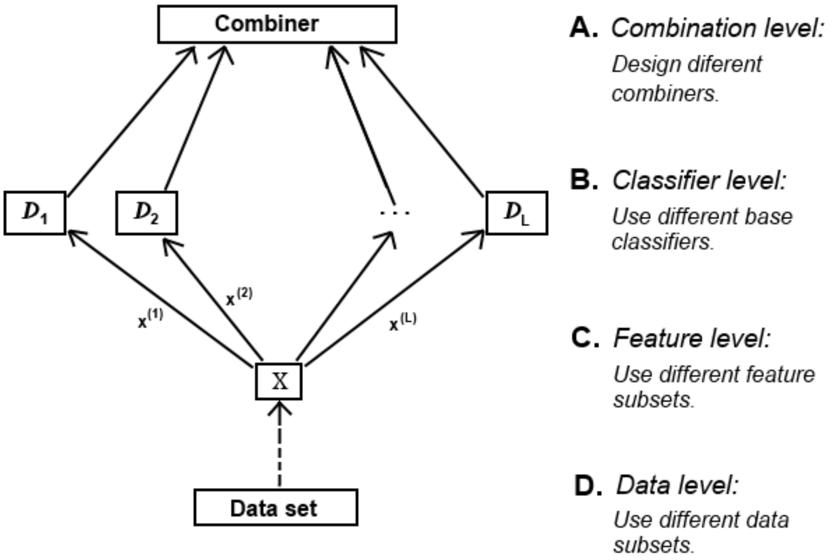

Fig. 3. Approaches to build classifier ensembles (Kuncheva, 2004).

5. Proposed Approach

The architecture of the proposed system is represented in Figure 4. It depicts the different components of the application and reveals the system's processing flow, since an input text is load to the application until it produces the desired output, i.e. a set of keywords for each of the unlabeled files. We describe below the main stages of the system.

5.1 Preprocessing

Preprocessing tasks are usually a prerequisite to text classification. The objective of this stage is cleaning the input text by eliminating unnecessary words and characters (KUSCO, KEA) or, just structuring the text correctly (CRF), for further classification.

KEA and KUSCO already perform preprocessing tasks internally. CRF requires extra/different preprocessing operations than those needed by the other two applications. In the latter, in order to identify most of the features, a slight modification in the text is needed. Two major aspects were then taken into account, during this phase:

Structural issues:

1.1.1. Separating each phrase of the text, one per line;
1.1.2. Keeping punctuations present in each phrase (except quote marks, which actually produced better results when removed);
1.1.3. Keeping stop-words present in each phrase;
1.1.4. Each token of each phrase separated by a space.

Feature and keyword automatic tagging:

1.1.5. Tagging the true keywords of each document within brackets (e.g.: [correctly tagged keyword]);
1.1.6. Identifying a set of features present in the text, namely those described in Table 1.

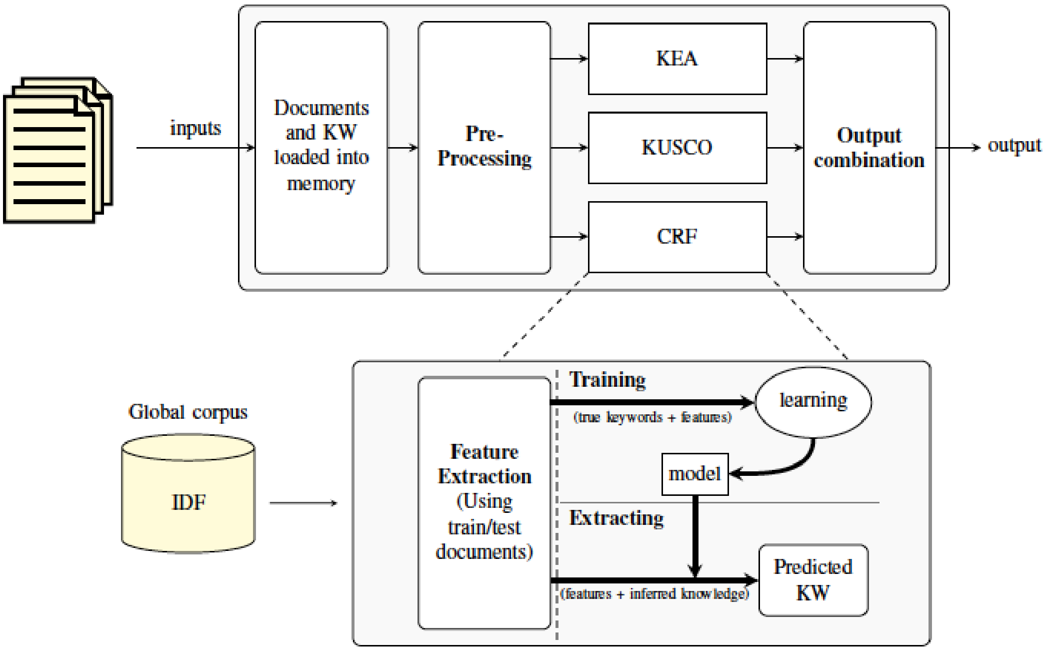
inputs
Documents and KW loaded into memory
Pre-Processing
KEA
KUSCO
CRF
Output combination
output
Global corpus
IDF
Feature Extraction (Using train/test documents)
Training
(true keywords + features)
learning
model
Extracting
(features + inferred knowledge)
Predicted KW

Fig. 4. Proposed system's architecture.

Features	Explanation	Range

1	Word	current token	-
2	PoS	Part-of-Speech tag of a token	{DT, VB, NN, (…)}
3	First Position	if a token is the first token in a sentence	{0, 1}
4	CAPITALIZED	if a token is capitalized	{0, 1}
5	Initial CAP	if a token begins with a capital	{0, 1}
6	Mixed CAPS	if a token contains both lower and upper cases	{0, 1}
7	Contains Digits	if a token contains digits	{0, 1}
8	All Digits	if a token is a number	{0, 1}
9	Hyphenated	if a token contains hyphens	{0, 1}
10	Dollar Sign	if a token contains the **$** sign	{0, 1}
11	Ends In Dot	if a token ends with a dot	{0, 1}
12	Lonely Initial	if the token is an initial (e.g.: P.)	{0, 1}
13	Single Char	if the token is a single char (letter, number, symbol)	{0, 1}
14	End Punctuation	if the token is sentence end punctuation	{0, 1}
15	Apostrophe	if the token contains an apostrophe (')	{0, 1}
16	Line Number	the line number of the current sentence	{1, 2, … N}
17	TF	Term Frequency of the term in the document	[0, 1]
18	IDF	Inverse Document Frequency of the term in Wikipedia global corpus	ℝ
19	TF*IDF	the Term-Frequency * Inverse Document Frequency of a term in the document	ℝ
20	Windowed Features	the PoS and Word features of the *first*, *first and second* and *first, second and third* tokens before and after the current token	-

Tab. 1. List of Features used to train the CRF model in this work.

5.2 Keyword Classifiers

After the preprocessing phase, there are two important phases within the system: (i) the training phase, where a set of manually tagged texts is used to create a model for each classifier; and (ii) the extracting phase, where each classifier is used to classify new texts. Ideally, these models should complement to one another, each one being specialized in a part of the domain where the others do not perform so well (just as human executives seek advisers whose skills complement each other). The rationale is to use two already existing applications (KEA and KUSKO) which will be combined with a third one (CRF). The latter is implemented with that purpose in mind making the overall proposed system efficient as will be demonstrated further in the paper. If we look at Figure 3, this corresponds to the classifier level (Approach B). Furthermore, the features taken into account by each system are different, so we are also introducing some diversity at the feature level (Approach C).

The CRF implementation used in this work was developed under MALLET[2] framework, acronym for MAchine Learning for LanguagE Toolkit. Table 1 describes the list of features used to train the CRF model. It also provides a short explanation of each one as well as their range domain.

Only in the case of KEA it was possible to define the specific number of keywords to be extracted. On the other hand, KUSCO and CRF may extract as much keywords as possible. As a result we set a threshold limiting the keywords that can be extracted. It was found that by setting a maximum of thirty keywords per application the results were the best.

While KEA and KUSCO are well known and have been described in sections 3.1 and 3.2, respectively; a zoomed view of the CRF, in particular, focusing on how it works, can also be gleaned from Figure 4.

5.3 Combine Classifiers

The output combination of the keyword classifiers corresponds to the ensemble level shown in Figure 4. It exploits the different ways of combining the outputs of *L* classifiers of an ensemble *D* depending on the type information is obtained from the individual members (Kuncheva, 2004). We review herein the majority voting method in view of the two most used versions: simple and weighted.

In simple Majority Voting (MV), as the name indicates, the consensus result is given as if a voting procedure takes place. Let *x* a feature vector, $\{\omega_1, \omega_2, \dots, \omega_c\}$ the set of *c* classes and $\{D_1, D_2, \dots, D_L\}$ the set of *L* classifiers. The output of the *i*th classifier is denoted by the *c*-dimensional binary vectors $D_i(x) = \left[d_{i,1}(x), \dots, d_{i,c}(x)\right]^T$, where $d_{i,1}(x) \in \{0,1\}$ is the degree of support given by classifier D_i to the assumption that *x* comes from class ω_1. We assume $d_{i,j} = 1$ if D_i categorizes/labels *x* in ω_j, and *0* otherwise. The plurality vote will result in an ensemble decision for class ω_k if

$$\sum_{i=1}^{L} d_{i,k} = \max_{1 \le j \le c} \sum_{i=1}^{L} d_{i,j} \tag{2}$$

The plurality vote of (2) can be written in a simpler way as $\sum_{i=1}^{L} d_{i,j} = \# \, [i\colon d_{i,j} = 1]$ = *number among the L voters that elected class j* and is the most used rule from the majority vote group.

In Weighted Majority Voting (WMV) more weight is given to the more competent classifiers thus strengthening their importance in making the final decision. The label outputs can be represented as degrees of support for the classes in the following way:

$$d_{i,j} = \begin{cases} 1, \text{ if } D_i \text{ categorizes } x \text{ in } \omega_j\,, \\ \quad 0, \text{ otherwise.} \end{cases} \tag{3}$$

The discriminant function for class ω_j obtained through weighted voting is

$$g_j(x) = \sum_{i=1}^{L} \omega_i d_{i,j}(x), \tag{4}$$

[2] MALLET (http://mallet.cs.umass.edu/) is a Java-based package for statistical natural language processing, document classification, clustering, topic modeling, information extraction, and other machine learning applications to text.

where ω_i is a coefficient which represents the weight for classifier D_i. Thus the value of the discriminant function $g_j(x)$ is the sum of the weights for those classifiers (of the ensemble) whose output for x is ω_j.

6. Experimental Setup

In this section we present the datasets, indicate the evaluation metrics and describe the main steps of the research design in order to evaluate the proposed approach.

6.1 Datasets

We describe in the sequel the four different datasets of English texts. The first dataset from now on addressed as *Hulth's dataset*, consists of *2000* scientific journal *paper abstracts* with their corresponding title, from the Inspec[3] database.

Hulth's documents were obtained from *Computers and Control* and *Information Technology* and have been widely used in previous related works (e.g. (Mihalcea *et al.*, 2004; Hulth, 2003). The second dataset, from now on designated by *Krap's dataset*, consists of 2304 *full papers* from *Computer Science* domain. Each document has clearly indicated its title, abstract, body and references (Krapivin *et al.*, 2009). Nevertheless, only the abstracts were used in this work. For validating the results obtained from the previous scientific datasets, two other collections of documents were used. Therefore, the third dataset is composed by 420 descriptions about *events* in general, like theatre plays and music concerts, retrieved from YourSingapore web-pages. The fourth one, comprises 112 *music personalities*' descriptions extracted from Wikipedia. The urban events collection was manual labelled for only one volunteer, while for the music personalities data set twenty volunteers were available for the labelling task.

6.2 Evaluation Metrics

For evaluating system performance, the already traditional four performance metrics from the Information Retrieval area were used: Precision (P), Recall (R) and *F-measure* (also known as F_1 score). In short, precision (P) represents the proportion of automatic selected keywords that are also manually assigned keywords, while recall (R) is the proportion of manually assigned keywords found by the automatic method. Thus, Precision may be seen as a measure of exactness whereas Recall represents a measure of completeness.

$$Precision = \frac{\#\ of\ manual\ keywords\ automatically\ selected}{Total\ \#\ of\ automatically\ selected\ terms} \tag{5}$$

[3] http://www.theiet.org/resources/inspec/ (last accessed 07/03/2016)

$$Recall = \frac{\#\ of\ manual\ keywords\ automatically\ selected}{\#\ of\ manual\ keywords\ in\ document} \tag{6}$$

The *F-measure* is the harmonic mean between precision and recall defined in equation 7:

$$F - measure = 2\ \times \frac{Recall\ \times Precision}{Recall + Precision} \tag{7}$$

In the great majority of the experiments herein presented, there is no particular reason to favor precision or recall, so we use equal weight of precision and recall to compute *F-measure*.

6.3 Research Design

We describe, in the following, 7 tests which run on the datasets described. The rationale is to empirically show the efficiency of the ensemble architecture proposed in this paper. Tests 1 to 5 were applied over abstracts from Hulth's dataset and only Test 5 was replicated on abstracts from Krap's dataset. For Krap's dataset, Tests 1 to 3 and 4 were not performed because as we extracted the title and abstract from the full documents, all the possible miss-indentations, miss-structuring and missing keywords were immediately corrected and the labelling method already used the stems rather than the full keywords. Tests 6 and 7 aimed at verifying whether the type and number of the gold standard keywords may influence the learning of the classifiers, filtering some of the documents out of the dataset.

The first test - Test 1 - can be seen as the baseline test. It was performed knowing in advance that only about 76% of the keywords were in fact present in the abstracts on the Hulth's dataset. This is explained by the author due to the fact that volunteers had access to full articles and not only to the abstracts in order to manually identify correct keywords (Hulth, 2004). Beyond that, this initial test was also conducted without any kind of extra preprocessing to the text files, i.e., documents are delivered to each of the applications without suffering any modifications.

The second test - Test 2 - differs from the first in one aspect: keywords that did not exist in abstracts were removed from the respective file's true keywords so, at this point, it is guaranteed that 100% of the keywords can be in fact found in the document they belong to. This was achieved using the keywords stems to find occurrences of each keyword. This test was performed in order to perceive how the true extraction ability of each classifier was.

In Test 3 we used the OpenNLP[4] Sentence Splitter to preprocess documents, structuring them so that each one contained only one sentence per line (that is to be tagged later). Despite no further enhancement is to be expected in KEA and KUSCO, once they do not have sentence structure into account, we might expect improvements from the CRF, since it uses many of the features from the text. In this regard, we clarify that some features to train our CRF model 2, like PoS tagging or Windowed Features, are most likely to suffer from badly structured sentences in documents (e.g. line breaks in the middle of sentences).

[4] http://opennlp.apache.org/

Until now, we automatically tagged true keywords exactly as they appeared in their respective .key files. In Test 4 we aim at verifying whether the method of labeling the true keywords (exact-keyword or stemmed-keyword) has impact in the learning procedure of the classifiers.

A closer look to the stemmer used (the English Porter Stemmer) showed that it was not performing as expected for some apparently basic cases. We found out that a new version of the stemmer, with some bugs corrected, was available to the community and instead of the old one we used - Test 5[5].

In Test 6 we removed the documents whose keywords contained digits. The rationale here was to understand if this type of keywords has impact in the classifiers' performance. Besides, the number of these documents was low (about 10% of the total number for Hulth's dataset, less than 10% for the others).

In Test 7, in addition to the constraint imposed in the previous test, only documents having between five and ten keywords were used. The reason is that while some documents had only one assigned keyword others had more than fifteen, i.e., the difference between minimum and maximum number of keywords was too big. It turned out that some training groups had considerably different average of keywords per file among them, making it harder to interpret the respective results.

7. Experimental Results and Discussion

In this section we present the results for keyword extraction, discuss the methodology and give evidence that our approach with a consensus of applications improves significantly the state of the art in identifying information related to textual document sources.

7.1 Results

For the performance evaluation of the ensemble classifiers we used ten-fold cross validation and selected the F_1-score as the metrics' indicator.

The experimental results are shown in Figures 5, 6 and 7, where the micro-averaged F_1 scores of each application are depicted for each dataset. Notice that the methods are indicated in the right by their acronyms, namely, KEA, KUSKO, CRF (classifiers) and MV, WMV1 and WMV2 (ensembles).

[5] The one available in http://snowball.tartarus.org/algorithms/english/stemmer.html.

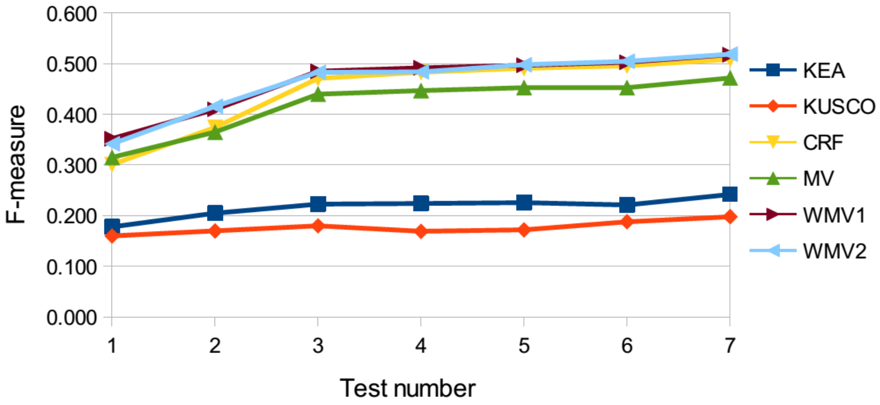

Fig. 5. Hulth's dataset results.

In the latter WMV1 and WMV2 denote the weight majority vote approach in two cases of F-measure as will be detailed in Table 2.

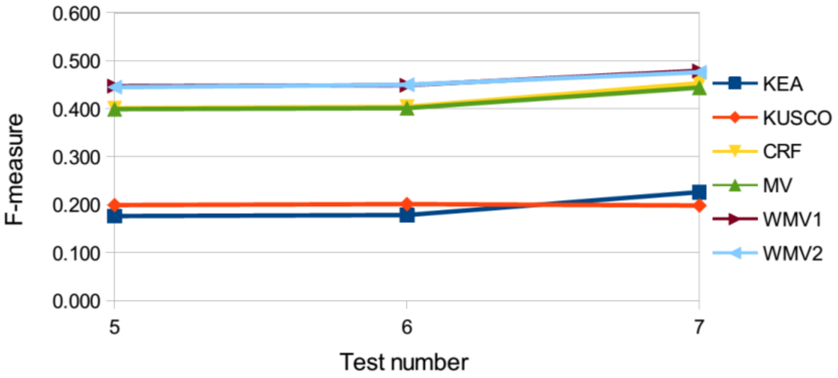

Fig. 6. Krap's dataset results.

Figure 7 illustrates the histogram of the best results obtained with the Hulth's, Krap's, Events and Personalities datasets.

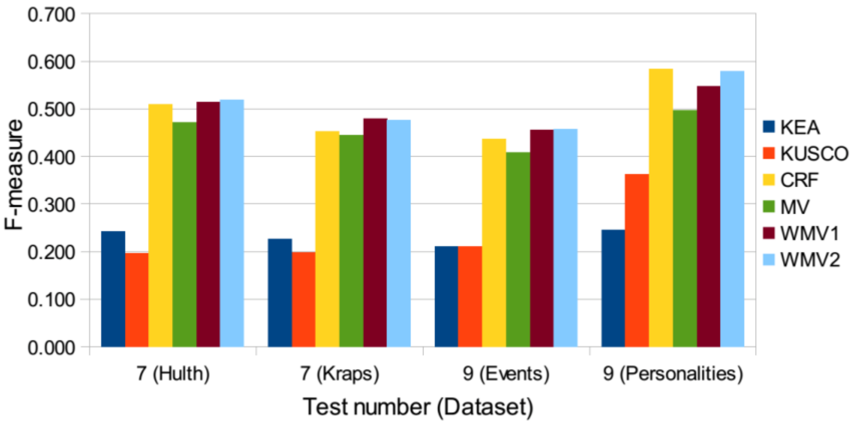

Fig. 7. Results validation.

The weight of each classifier in the final ensemble can be learned from their individual performance. Taking into account the number of keywords found by a classifier compared to the number of true keywords present in each text in a given labeled dataset, several Regression algorithms in Weka framework[6], were used with Pace Regression algorithm yielding the best result with a precision of 0.65 ± 0.13.

In Table 2 we present two different weight configurations (WMV1 and WMV2) that yielded to the best results, with Weighted Majority Voting. Each row corresponds to the weight configuration depending on how well CRF performed.

Weighted Ensemble	**CRF weight**	**KUSCO weight**	**KEA weight**
WMV 1 (CRF $F_1 \leq$ 0.48)	54%	36%	10%
WMV 2 (CRF $F_1 >$ 0.48)	62%	30%	8%

Tab. 2. Model weights producing best results.

7.2 Discussion

The attentive reader will notice that despite KEA is achieving better F_1 scores than KUSCO in almost all tests shown, it is always given less vote weight. This can be explained due to the type of the extracted keywords: KUSCO guarantees that each term it extracts is unique while KEA does not guarantee that. In fact, many of the terms extracted by KEA contain each other (e.g. [extracted example keyword], [extracted keyword], [example keyword]), which give an undesired emphasis to the *same* keyword when the voting phase occurs. To avoid this, the vote weight of KEA had to be lower than that of KUSCO.

[6] Weka is a collection of machine learning algorithms for data mining tasks, available at http://www.cs.waikato.ac.nz/ml/weka/.

During the experimentation we noticed that CRF yielded higher Precision (the keywords actually found) than Recall (the number of keywords found). This means that keywords extracted by CRF are usually correct but are generally not many, so the improvement that we observe in the final results (for the ensemble) come from compensating this lack of keywords that CRF can extract for some documents with the other applications.

Document structure was another factor influencing the results obtained. As one can see, after applying the OpenNLP Sentence Splitter to split the sentences of each document correctly, great improvement was observed. This happened because features as PoS and Windowed Features used to train CRF became more effective, since they depend on the structure of the sentence being analysed.

Yet concerning structural issues, the number of true keywords present in the files seems to affect the performance of the applications, as well as the type (precisely in this case, if they contained digits) of keywords given as gold standard. Thus, removing unseen keywords and those that contain numbers resulted in better performances observed.

It can also be observed from the results, since Test 2, that the Majority Voting (MV) no longer improves the results of the best individual application (CRF). This can be explained due to the huge differences in the classifiers' reliability, which differs much from one application to the other. Nevertheless, CRF takes advantage of the other classifiers in the ensemble like KUSCO specifically because the latter can cover a great diversity of simple and compound keywords since it uses WordNet and Wikipedia for keyword verification. This is done without the need of training neither labelled examples.

Although the results obtained with the baseline sources (scientific datasets) were good, to validate our approach we tried out with the urban events collection and music's personalities. From the results concerning those of the urban events descriptions, a performance improvement has been achieved, similarly to what happened with the scientific datasets. Nevertheless, for the music personalities' descriptions, the results using the ensemble did not improve those obtained by the CRF itself, even with the different configuration used for the Weighted Majority Vote. This can be explained because the high performance attained by CRF with this dataset (see figure 7) indicates that above a certain threshold, further gains are out of reach. In fact, the difference between the classifiers' performances is too large, and any keyword obtained with both classifiers (KUSCO and KEA) is not enough to improve the performance as compared to CRF. The results presented also seem to attest that CRF is highly dependent on the document structure and its type. Moreover, all the documents pertaining to the music personalities' dataset are very similar among each other, which seem to be the reason why CRF achieves even better performance in this case.

7.3 Statistical Evidence of Results

Table 3 summarizes the statistical significance of the difference between 6 methods by means of t-test. The significance level is set as 5%, so that the p-value less than 5% indicates that the two underlying methods are significantly different in the mean. As it may be observed, the methods CRF, MV, WMV1 and WMV2 significantly outperform the KEA and KUSKO methods in terms of F_1-score, as shown by the low p-values indicating the t-test is highly significant at significance levels of 1%. There is no statistical evidence that any of the ensemble approaches performs better than the CRF. The p-value 0.023495 shows that WMV2 is significantly different in the mean at the level of 5% with respect to the Majority

Voting (MV). Also the p-value 0.005822 shows that the difference between the means of WMV1 and MV is statistically significant.

	b	**c**	**d**	**e**	**f**
a	0.758716	0.001968**	0.000484**	0.000360**	0.000854**
b		0.001854**	0.009524**	0.004325**	0.002573**
c			0.082666	0.924491	0.280540
d				0.005822**	0.023495*
e					0.331080

Tab. 3. Significance Tests with statistical variable F1-score: (a) KEA (b) KUSKO (c) CRF (d) MV (e) WMV1 (f) WMV2. The symbol * indicates that the test is significant at 5% level; ** means highly significant at 1% level.

8. Conclusions and Future Work

The work presented in this paper exploits AKE from textual sources in general, since it was successfully applied to scientific and non-scientific domains.

The proposed approach builds a consensus-based machine learning methodology (both supervised and unsupervised). Moreover, using an ensemble of several applications to improve the performance revealed to be very effective over single classifiers results. For combining the outputs of the individual models, two methods of majority voting are used: simple majority and weighted majority. While the former gives equal weight to all predictive models, the latter gives more weight to those who present better predictive performance. However, one factor that can limit the enhancement seems to be the difference in each application's performance: the more one of the applications outperforms the others, the smaller are the gains.

This work has shown that by combining models of different existing applications, instead of using a more traditional method to generate different models, is also a viable method to create an ensemble application and the empirical results here obtained, which improved those of each the individual systems, attest that.

In future work it will be worth improving the CRF itself, by adding new features that were not present yet but might be considered important. Indeed, we will apply the learned system to urban event collections from other English-native cities as Boston and London in order to verify the generality of our approach independently of regional influences on text styles. The resultant keywords extracted will be used to improve predicting models on transport demand due people presence in the urban area on these cities.

9. Acknowledgements

This work was supported by the InfoCrowds project-PTDC/ECM-TRA/1898/2012 and by the REMINDS project- UTAP-ICDT/EEI-CTP/0022/2014. It also involved the collaboration with the Massachusetts Institute of Technology (MIT).

We gratefully acknowledge Anette Hulth for pointing us toward valuable scientific datasets and to Su Nam Kim for maintaining the public on-line repository.

10. References

Alves, A., Antunes, B., Pereira, F., and Bento, C., 2009. Semantic Enrichment of Places: Ontology Learning from Web, Int. J. Know.-Based Intell. Eng. Syst. 13 (1) 19-30. http://dx.doi.org/10.3233/KES-2009-0170

Brill, E., 1994. Some Advances in Transformation-Based Part of Speech Tagging, in: In Proceedings of the Twelfth National Conference on Articial Intelligence, 722-727.

Escovedo, T., Cruz, A., Koshiyama, A., Melo, R., Vellasco, M., 2014. NEVE++: A Neuro-Evolutionary Unlimited Ensemble for Adaptive Learning, in: Proceedings of the International Joint Conference on Neural Networks, IJCNN '14, Beijing, China, 3331-3338.

Finkel, J., Grenager, T., Manning, C., 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling, in: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL '05, Association for Computational Linguistics, Stroudsburg, PA, USA, 363-370. http://dx.doi.org/10.3115/1219840.1219885

Grineva, M., Grinev, M., and Lizorkin, D., 2009. Extracting Key Terms from Noisy and Multitheme Documents, in: Proceedings of the 18th International Conference on World Wide Web, WWW '09, ACM, New York, NY, USA, 661-670. http://dx.doi.org/10.1145/1526709.1526798

Hulth, A., 2008. Automatic Keyword Extraction: Combining Machine Learning and Natural Language Processing, VDM Verlag, Saarbrücken, Germany.

Hulth, A, 2003. Improved Automatic Keyword Extraction Given More Linguistic Knowledge, in: Proceedings of the 2003 Conference on EmpiricalMethods in Natural Language Processing, Empirical Methods in NLP, Association for Computational Linguistics, Stroudsburg, PA, USA, 216-223. http://dx.doi.org/10.3115/1119355.1119383

Kim, S., Medelyan, O., Kan, M.-Y., and Baldwin, T., 2010. SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientic Articles, in: Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval '10, Association for Computational Linguistics, Stroudsburg, PA, USA, 21-26.

Krapivin, M., Autaeu, A., Marchese, M., 2009. Large dataset for keyphrases extraction, Tech. Rep. Tech. Report DISI-09-055, University of Trento, Italy.

Kuncheva, L., 2004. Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience. http://dx.doi.org/10.1002/0471660264

Laerty, J., McCallum, A., and Pereira, F., 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, in: Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 282-289.

Li, Z., Zhou, D., Juan, Y.-F., and Han, J., 2010. Keyword Extraction for Social Snippets, in: Proceedings of the 19th International Conference onWorld Wide Web, WWW '10, ACM, New York, NY, USA, 1143-1144.

Luhn, H., 1958. The automatic creation of literature abstracts, IBM J. Res. Dev. 2, 159-165. http://dx.doi.org/10.1147/rd.22.0159

Matsuo, Y., and Ishizuka, M., 2004. Keyword Extraction from a Single Document using Word Co-Occurrence Statistical Information, International Journal on Articial Intelligence Tools 13 (1) 157-169. http://dx.doi.org/10.1142/S0218213004001466

Medelyan, O., Frank, E., and Witten, I., 2009. Human-competitive Tagging Using Automatic Keyphrase Extraction, in: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3, EMNLP '09, Association for Computational Linguistics, Stroudsburg, PA, USA, 1318-1327.

Medelyan, O., and Witten, I., 2006. Thesaurus Based Automatic Keyphrase Indexing, in: Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL '06, ACM, New York, NY, USA, 296-297. http://dx.doi.org/10.1145/1141753.1141819

Mihalcea, R., and Tarau, P., 2004. Textrank: Bringing order into textl, in: Conference on Empirical Methods in Natural Language Processing (EMNLP), 404-411.

Plas, L., Pallotta, V., Rajman, M., Ghorbel, H, 2004. Automatic Keyword Extraction from Spoken Text. A Comparison of two Lexical Resources: the EDR and WordNet, CoRR cs.CL/0410062.

Peng, F., and McCallum, A., 2006. Information Extraction from Research Papers Using Conditional Random Fields, Inf. Process. Manage. 42 (4) 963-979. http://dx.doi.org/10.1016/j.ipm.2005.09.002

Ramshaw, L., and Marcus, M., 1999. Text Chunking using Transformation-Based Learning, CoRR .

Robertson, S., 2004. Understanding inverse document frequency: On theoretical arguments for IDF, Journal of Documentation 60, 503-520. http://dx.doi.org/10.1108/00220410410560582

Rose, S., Engel, D., Cramer, N., Cowley, W., 2010. Automatic Keyword Extraction from Individual Documents, in: M. W. Berry, J. Kogan (Eds.), Text Mining. Applications and Theory, John Wiley and Sons, Ltd, 1-20. http://dx.doi.org/10.1002/9780470689646.ch1

Sarkar, K., Nasipuri, M., and Ghose, S., 2010. A New Approach to Keyphrase Extraction Using Neural Networks, CoRR abs/1004.3274.

Sutton, C., and McCallum, A., 2007. An Introduction to Conditional Random Fields for Relational Learning, in: L. Getoor, B. Taskar (Eds.), Introduction to Statistical Relational Learning, MIT Press.

Timonen, M., Toivanen, T., Teng, Y., Chen, C., and He, L., 2012. Informativeness-based Keyword Extraction from Short Documents, in: A. L. N. Fred, J. Filipe, A. L. N. Fred, J. Filipe (Eds.), KDIR, SciTePress, 411-421.

Turney, P., 2000. Learning Algorithms for Keyphrase Extraction, Information Retrieval 2 (4) 303-336, ISSN 1386-4564.

Wan, X., and Xiao, J., 2008. Single Document Keyphrase Extraction Using Neighborhood Knowledge, in: AAAI, 855-860.

Wang, J., Peng, H., Hu, J.-S., and Zhang, J., 2006. Ensemble learning for keyphrases extraction from scientic document, in: J. Wang, Z. Yi, J. M. Zurada, B.-L. Lu, H. Yin (Eds.), ISNN'06, Lecture Notes in Computer Science, Springer, Berlin, 1267-1272.

Witten, I., Paynter, G., Frank, E., Gutwin, C., and Nevill-Manning, C., 1999. KEA: Practical Automatic Keyphrase Extraction, in: Proceedings of the Fourth ACM Conference on Digital Libraries, DL '99, ACM, New York, NY, USA, 254-255. http://dx.doi.org/10.1145/313238.313437

Yu, F., Xuan, H., and Zheng, D., 2012. Key-Phrase Extraction Based on a Combination of CRF Model with Document Structure, in: 8th Int. Conference on Computational Intelligence and Security, IEEE, 406-410. http://dx.doi.org/10.1109/cis.2012.97

Zhang, C., 2009. Combining Statistical Machine Learning Models to Extract Keywords from Chinese Documents, in: R. Huang, Q. Yang, J. Pei, J. Gama, X. Meng, X. Li (Eds.), Advanced Data Mining and Applications, vol. 5678 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 745-754. http://dx.doi.org/10.1007/978-3-642-03348-3_79

Zhang, C., Wang, H., Liu, Y., Wu, D., Liao, Y., and Wang, B., 2008. Automatic keyword extraction from documents using conditional random fields, Journal of Computational Information Systems 4 (3) 1169-1180.