SemGraph: Extracting Keyphrases Following a Novel Semantic Graph-Based Approach

Juan Martinez-Romo
NLP & IR Group, Dpto. Lenguajes y Sistemas Informáticos, Universidad Nacional de Educación a Distancia (UNED), Juan del Rosal, 16. 28040 Madrid, Spain. E-mail: juaner@lsi.uned.es

Lourdes Araujo
NLP & IR Group, Dpto. Lenguajes y Sistemas Informáticos, Universidad Nacional de Educación a Distancia (UNED), Juan del Rosal, 16. 28040 Madrid, Spain. E-mail: lurdes@lsi.uned.es

Andres Duque Fernandez
NLP & IR Group, Dpto. Lenguajes y Sistemas Informáticos, Universidad Nacional de Educación a Distancia (UNED), Juan del Rosal, 16. 28040 Madrid, Spain. E-mail: aduque@lsi.uned.es

Keyphrases represent the main topics a text is about. In this article, we introduce SemGraph, an unsupervised algorithm for extracting keyphrases from a collection of texts based on a semantic relationship graph. The main novelty of this algorithm is its ability to identify semantic relationships between words whose presence is statistically significant. Our method constructs a co-occurrence graph in which words appearing in the same document are linked, provided their presence in the collection is statistically significant with respect to a null model. Furthermore, the graph obtained is enriched with information from WordNet. We have used the most recent and standardized benchmark to evaluate the system ability to detect the keyphrases that are part of the text. The result is a method that achieves an improvement of 5.3% and 7.28% in F measure over the two labeled sets of keyphrases used in the evaluation of SemEval-2010.

Introduction

Given the large number of documents available for any topic and task, it is very useful to have access to a set of accurate descriptors that help us to select those we actually need. Automatic summarization is aimed at any type of document, including scientific papers, Web pages, and news, with the purpose of generating the summary information expressed in a reduced set of terms. A particular type of summarization often addressed in the literature is keyphrase extraction, where the goal is to select a set of phrases to represent a document. Keyphrases are a sequence of one or more words that capture the main topics of a document. They may or may not appear in the text, though our systems only look for those appearing in the text.

Accordingly, this task is a useful utility for many text-mining and document classification applications.

Automatic keyphrase extraction systems are divided into two groups, depending on their approach to the problem: supervised or unsupervised. Supervised systems generally extract a set of features to represent each document in a training collection. This representation is then used to learn a model with a classification algorithm, produce an inferred function, and predict new instances from a test collection. Some of the main drawbacks of such systems are that they need a training data set typically labeled by humans, and also the learning algorithm requires to generalize from the training data to unseen situations. Unsupervised systems deal with the problem of trying to find hidden structure in unlabeled data. Given the facts that human labeling is a time-consuming task, and that unsupervised learning allows to learn larger, more-complex models than supervised learning, we present, in this study, an unsupervised keyphrase extraction system.

Our Approach

We propose a new approach for the selection of candidate terms, taking into account their semantic relationships. Our approach to the problem relies on a semantic relationship graph differing from other approaches in relevant aspects.

Received January 29, 2014; revised June 9, 2014; accepted June 10, 2014

© 2015 ASIS&T • Published online 5 May 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/asi.23365

We assume that a document is a coherent piece of information, and thus words in a document tend to (statistically) adopt a related sense. This hypothesis is supported by previous works, such as the study by Gale, Church, and Yarowsky (1992), which showed that if a polysemous word appears two or more times in a discourse, it is extremely likely that all uses will share the same sense. For this assumption to make sense, we take as a document the set of sections usually related to the main topic in research articles. In addition, keywords tend to appear in particular sections of the articles, and it is common, in other works that deal with keyphrase identification, to consider only specific sections.

Whereas previous proposals connect two words if they co-occur within a window of a maximum number of words, we consider the whole document to be a coherent piece of information. Our aim is to create a link joining every two words adopting related senses, so co-occurrence in the same document will be taken as a proxy for this. However, we know this is not strictly true. Some words may appear in a document without really being related to its main topic meaning. So, we can only consider that two words truly adopt related senses if they are *often* found together in the same documents.

SemGraph, based on the previous hypothesis, uses the statistical significance respect to a null model to create a single semantic relationship graph out of the whole corpus, in a similar way as it is done in a previous work (Martinez-Romo et al., 2011), but with a different purpose. Thus, we consider that SemGraph contains semantic information for two reasons. First, relations between terms are not based on a simple co-occurrence, but their relationships are established as a result of a significant number of occurrences within a document that represents a semantic unit. Second, the graph obtained is enriched with semantic information from WordNet.

Previous work (Martinez-Romo et al., 2011) is a theoretical article that describes the building of the graph used for the extraction of semantic relationships introduced in this article. The extraction of keyphrases is a problem that can benefit from graph techniques because helps to select relevant information from a whole piece of information (section, document, and so on). As for the other aspects involved in our algorithm, this graph allows to carry out a more efficient term weighting than the frequency-based methods used until now in most systems. In addition, we carry out an analysis of the logical structure of each document. Because we are working with scientific articles, the main aim of this analysis is to extract keywords only from the most relevant sections, that is, from those sections that usually contain the keyphrases.

The remaining of the paper proceeds as follows: Related Work presents the related work; Algorithm Description is devoted to the algorithm description; System Performance Analysis provides an analysis of system performance; Experimental Results exposes the most significant results; finally, Conclusions and Future Work draws the main conclusions and future work.

Related Work

There are two main approaches to extracting keyphrases: supervised and unsupervised. The first proposals addressing the problem of extracting keyphrases were based on supervised machine-learning algorithms (Turney, 2000; Witten, Paynter, Frank, Gutwin, & Nevill-Manning, 1999). KEA (Witten et al., 1999) identified candidate keyphrases using lexical methods, then calculated feature values for each candidate, and used a machine-learning algorithm to select which candidates were considered keyphrases.

This machine-learning scheme builds a prediction model using training documents with known keyphrases and then uses the model to find keyphrases in new documents. The candidate phrases are selected by applying a set of simple rules, namely, candidate phrases are limited to a certain maximum length (usually three words), candidate phrases cannot be proper names, and candidate phrases cannot begin or end with a stopword. Among the features considered for the machine-learning algorithm were Tf-Idf (term frequency–inverse document frequency) and the position of the first occurrence of the candidate phrase. KEA used the Naïve Bayes technique as a training algorithm. Turney (2000) introduced a larger set of features, including the number of words in the phrase, word frequencies, and the presence of proper nouns, among others. Candidate terms were all stemmed unigrams, bigrams, and trigrams from the documents, after stopword removal. Turney reported that a genetic algorithm specifically designed for the task identifies better keywords than the general-purpose C4.5 decision-tree–induction algorithm. Later, Hulth (2003), following also a machine-learning approach, found that extracting NP-chunks gives better precision than n-grams, and by adding the POS tag(s) assigned to the term as a feature, an important improvement of the results is obtained. She used four features: term frequency, collection frequency, relative position of the first occurrence, and the POS tag(s) assigned to the term. The machine-learning approach used in this work was rule induction, where the rules were constructed by recursively partitioning the classes for each rule. A difference between this study and the prior work is that this work is focused on keyword extraction from abstracts, instead of full-length texts.

Another machine-learning proposal (Wang, Mu, & Fang, 2008) introduced semantic information as a new feature in the machine-learning scheme. One of these features took into account the semantic similarity between words computed from WordNet. Results were superior to those obtained using KEA.

Unsupervised Systems

Given that SemGraph follows an unsupervised approach, we analyze works close to our research. Mihalcea and Tarau (2004) proposed TextRank, a graph-based ranking model for term selection. They constructed a graph among terms considering co-occurrence relations, controlled by the distance between word occurrences: Two vertices are connected if their corresponding lexical units co-occur within a window of a maximum number of words (between 2 and 10). After the undirected unweighted graph is constructed, the score associated with each vertex is set to an initial value of 1, and a ranking algorithm is applied. Then, they apply a graph-based ranking algorithm similar to PageRank for deciding the importance of a vertex within a graph, based on global information recursively drawn from the entire graph. Once a final score is obtained for each vertex in the graph, vertices are sorted in reverse order of their score, and the top *T* vertices in the ranking are retained for postprocessing, where *T* is set to one third of the number of vertices in the graph. During postprocessing, all lexical units selected as potential keywords by the TextRank algorithm are marked in the text, and sequences of adjacent keywords are collapsed into a multiword keyword. They used the same test data as Hulth (2003), but, in this case, the separation in test and training data was not needed. TextRank achieved the highest precision and F measure across all systems participants in the Document Understanding Conference 2002 (DUC, 2002) competition, although the recall was not as high as in supervised methods.

Liu, Li, Zheng, and Sun (2009b) also followed an unsupervised approach. They tried to improve the keyphrases coverage of the document content by applying clustering techniques to select the candidate terms. First, the terms in a document are grouped into clusters according to some measure of semantic distance. Each cluster is represented by a centroid term. Then, the keyphrases are extracted from the document using these exemplar terms. This is done by extracting the noun groups whose pattern is zero or more adjectives followed by one or more nouns. This method outperformed the F measure of TextRank.

In a later work, Liu, Huang, Zheng, and Sun (2010) pursued a similar goal, proposing a topic decomposition approach aiming to obtain the importance scores of words under different topics. Once the document topics have been obtained, they extract keyphrases that are relevant to the document and, at the same time, have a good coverage of the document's major topics. They used Wikipedia to infer latent topics with latent Dirichlet allocation (Blei, Ng, & Jordan, 2003), taking each Wikipedia article as a document. Results slightly improved those of TextRank.

Hasan and Ng (2010, p. 372) conducted a comparative study of the different unsupervised proposals, including the ones mentioned above, applying them to new corpora. Unfortunately, the results were not conclusive becasue the systems that worked best in a corpus were among the worst for others. However, they provide an interesting observation, namely, that "the way of forming phrases, rather than the window size, has the largest impact on the scores."

Qazvinian, Radev, and Ozgur (2010) presented an approach to summarize single scientific articles by extracting its contributions from the set of citation sentences written in other articles. The summarization is based on extracting significant keyphrases from the set of citation sentences and using these keyphrases to build the summary. In this setting, the best summary would have as few sentences and, at the same time, as many keyphrases as possible.

Lopez, Barreda, Tejada, and Cuadros (2011) presented an unsupervised graph-based method to extract keyphrases using MFS (Maximal Frequent Sequences) and building the nodes of a graph with them. The weight of the connection between two nodes has been established according to common statistical information and semantic relatedness.

Newman, Koilada, Lau, and Baldwin (2012) proposed an unsupervised model called Dirichlet process segmentation for identifying correct spans in index term and keyphrase extraction. The experimental results showed that this method obtained very competitive results, despite not being a system built specifically for the task.

Bougouin, Boudin, and Daille (2013) introduced TopicRank, a graph-based keyphrase extraction method that relies on a topical representation of the document. Candidate keyphrases are clustered into topics and used as vertices in a complete graph. A graph-based ranking model is applied to assign a significance score to each topic. Keyphrases are then generated by selecting a candidate from each of the top-ranked topics. The researchers compared TopicRank with previous work on four data sets, obtaining remarkable results.

Algorithm Description

This section describes the proposed algorithm for extracting keyphrases based on the semantic relationship graph. Our method roughly consists of four main steps:

1. ***Preprocessing.*** First, a cleaning of the text and an analysis of the logical structure of each document are carried out.
2. ***POS Tagging.*** A part-of-speech (POS; also called grammatical) tagging of the preprocessed documents is performed.
3. ***Semantic Relationship Graph.*** Previously POS tagged documents are used to build a semantic relationship graph.
4. ***Extraction and Selection of Keyphrases.*** Finally, the semantic relationship graph helps us to extract and select keyphrases.

Preprocessing and POS tagging steps are common in most keyphrase extraction works; thus, the stages that really characterize our novel method are the building of the semantic graph and the extraction and selection of keyphrases.

Preprocessing

The first stage of the keyphrase extraction process begins by cleaning the text that our system will analyze. Hence, result tables, mathematical formulas, author affiliations, and other messy lines are eliminated.

The second stage in the preprocessing is an analysis of the logical structure of each document. Several studies

(Lopez & Romary, 2010; Nguyen & Luong, 2010; Treeratpituk, Teregowda, Huang, & Giles, 2010) in the literature have shown that the analysis of this structure is a determining factor when extracting quality keyphrases. Because we are working with scientific articles, this logical structure could be formed by elements such as title, authors, abstract, and other typical sections. Among all the parts of a scientific article, there are some sections (Treeratpituk et al., 2010) whose relevance is higher than others in order to extract keyphrases from them.

The differences in the relevance of the sentences coming from different parts of the structure of the scientific articles have been made clear in previous work, such as that of Teufel and Moens (1998).

Thus, we have developed a program able to remove the following parts: authors, affiliation, acknowledgment, and references. It also detects the following sections: title, abstract, introduction, related work, future work, and conclusions. Some scientific articles do not present any of these sections or are named differently. For instance, the related work section is sometimes referred to as previous work or background. In the case of variations on the name, we have established a set of rules to detect such sections in each particular case. Title, abstract, introduction, and conclusions are detected in all items, whereas related work and future work may or may not be detected, depending on whether they have separate sections or form part of the introduction or conclusions, respectively. We decided to select these sections (title, abstract, introduction, related work, conclusions, and future work) and ignore the rest mainly for two reasons: the difficulty of detecting other sections, such as methodology or experiments, given its great ambiguity, and the low number of keyphrases they usually include.

Furthermore, the graph, collection, and keyphrases used in this algorithm have worked with stemmed terms. For that, we have used the Porter stemming algorithm implemented in the Snowball[1] library.

POS Tagging Documents

After the preprocessing of documents, we carry out a process of POS tagging. We annotate the documents with POS tags using the Stanford Log-linear POS Tagger.[2] In Hulth (2003), a study was conducted about how most keyphrases are usually composed of nouns and adjectives. Some works (Liu et al., 2009b) used regular expressions to extract keyphrases based on combinations of nouns and adjectives. These regular expressions focus on patterns formed by adjectives in the first part of the phrase (optionally), followed by unique combinations of singular, plurals, or proper names. Other works (Berend & Farkas, 2010) have considered adjectives, nouns, and verbs as the unique possible tokens in phrase candidates to represent a document. In addition, these phrases cannot begin or end with a stopword.

Our method is less restrictive, because it selects any noun or adjective, as in other works (Mihalcea & Tarau, 2004; Wan & Xiao, 2008b), because much of the weight of the keyphrase selection lies in the use of a semantic relationship graph. For the experiments, we selected as candidate tokens those labeled according to the Penn Treebank Tag Set[3] as: NN, NNS, NNP, NNPS, JJ, JJR, and JJS.

Building the Semantic Relationship Graph

As stated above, we assume that the appearance of a word in a document is usually related to the document topic, but not necessarily, because a word may appear by pure chance in a document. In this case, the co-occurrence of this word with the words actually related to the document topic is not expected to be significant.

Thus, co-occurrence is considered only if it is statistically significant, and this significance is reflected in the link strength. Accordingly, we need to assign a significance to the co-occurrence of two terms in a certain number of documents out of the whole collection. This is akin to statistical hypothesis testing, the hypothesis being that the two words co-occur because of semantic relatedness. Statistical hypothesis testing relies on the setting of a null model that defines what we consider pure chance. In our null model, elements are randomly and independently distributed among the documents of the collection. Co-occurrence will be considered statistically significant if it is unlikely that it arises by pure chance that is, generated by the null model. If two words are found respectively in n_1 and n_2 documents out of the N that form the corpus, to count in how many arrangements of two words coincide in exactly k documents we must realize that there are four sets of documents: k documents containing both words; $n_1 - k$ documents containing only the first word; $n_2 - k$ documents containing only the second word; and $N - n_1 - n_2 + k$ documents (provided this number is nonzero) containing none of the words. Thus, the sought number of arrangements will be given by the multinomial coefficient, as shown by Equation 1:

$$\binom{N}{k, n_1 - k, n_2 - k}. \tag{1}$$

Hence, the probability that two words that appear in n_1 and n_2 documents each and are randomly and independently distributed among N documents coincide in exactly k of them is obtained as shown by Equation 2:

$$p(k) = \binom{N}{n_1}^{-1} \binom{N}{n_2}^{-1} \binom{N}{k, n_1 - k, n_2 - k} \tag{2}$$

if $\max\{0,\ n_1 + n_2 - N\} \le k \le \min\{n_1,\ n_2\}$ and is zero otherwise.

[1] http://snowball.tartarus.org/

[2] http://nlp.stanford.edu/software/tagger.shtml

[3] http://www.cis.upenn.edu/~treebank/

We can write Equation 2 in a more convenient form to make it computationally practical. For that purpose, we introduce the notation $(a)_b \equiv a(a-1)\ldots(a-b+1)$, for any $a \geq b$, and without loss of generality assume that the first word is the most frequent word (i.e., $n_1 \geq n_2 \geq k$). Then (Equation 3)

$$\begin{aligned} p(k) &= \frac{(n_1)_k (n_2)_k (N-n_1)_{n_2-k}}{(N)_{n_2}(k)_k} \\ &= \frac{(n_1)_k (n_2)_k (N-n_1)_{n_2-k}}{(N)_{n_2-k}(N-n_2+k)_k (k)_k}, \end{aligned} \tag{3}$$

Where, in the second form, we have used the identity $(a)_b = (a)_c(a-c)_{b-c}$ valid for $a \geq b \geq c$. Equation 3 is better written as Equation 4:

$$p(k) = \prod_{j=0}^{n_2-k-1}\left(1-\frac{n_1}{N-j}\right) \times \prod_{j=0}^{k-1}\frac{(n_1-j)(n_2-j)}{(N-n_2+k-j)(k-j)}. \tag{4}$$

This allows us to determine a p value for co-occurrence of the two words as shown by Equation 5:

$$p = \sum_{k \geq r}^{n_2} p(k), \tag{5}$$

where r is the number of documents in the corpus where the two words are actually found together. If $p \lll 1$, we can consider that the appearance of the two words in the same document is significant, and therefore it is likely that their meaning is related. We can further quantify this significance by taking the median (corresponding to $p = 1/2$) as a reference and computing the weight of a link as $\ell = -\log(2p)$, that is, a measure of how much the actual value of r deviates from the median.

Then, we construct a graph from a set of semantic relationships with the words as nodes and joining with a link every two words that appear in at least one common document. A co-occurrence weight above a threshold value is assigned to each link. The weight ℓ assigned to the links measures the deviation of co-occurrence of the two words with respect to the null case. The resulting graph will henceforth be referred to as the *semantic relationship graph*.

Semantic enrichment. The second phase of construction of the graph is to enrich the semantic relationships between terms with information extracted from WordNet. WordNet (Fellbaum, 2010) is a freely available lexical database of English, which groups nouns, verbs, adjectives, and adverbs into sets of synonyms, each expressing a distinct concept called synset. The synsets (concepts) in WordNet are interlinked with conceptual-semantic and lexical relations.

A large number of semantic distance measures based on information found in the lexical database WordNet have been proposed (Agirre, de Lacalle, & Soroa, 2013; Banerjee & Pedersen, 2003; Hirst & St-Onge, 1997; Jiang & Conrath, 1997; Leacock & Chodorow, 1998; Lin, 1998; Patwardhan & Pedersen, 2006; Resnik, 1995; Wu & Palmer, 1994). Some of these proposals (Agirre et al., 2013; Banerjee & Pedersen, 2003; Hirst & St-Onge, 1997) ignore the directionality of relations, as it happens in our case.

First, we search for a correspondence between each node in the graph with one or more concepts in WordNet. We do not apply word-sense disambiguation. Therefore, a word in the graph can be represented by several synsets sharing the same lexical category in WordNet. Then, each relation between concepts is depicted in the graph as follows. Let $G = (V, E)$ be the co-occurrence graph, where V is the set of nodes and E the edges. Let W be the set of nodes in the graph that have a match on WordNet, so that $W \subset V$. Let R be the set of relationships between concepts in WordNet. Thus, if we take a node $v_i \in V$ and two nodes $w_i, w_j \in W$ between which there is a relationship $r_i(w_i, w_j) \in R$, and it holds that $e_i(w_i, v_i) \in E$, and $e_j(w_j, v_i) \notin E$, then we create a new edge in the graph $e_j(w_j, v_i)$ with the same weight as $e_i(w_i, v_i)$.

We ignore the relation type between two concepts. If two WordNet relations exist between two nodes, we only represent one edge and ignore the type of the relation. We chose to use undirected relations between concepts because most of the relations are symmetric, and in previous works (Agirre et al., 2013) selecting relations as directed links had negative effects.

Significance threshold. We have analyzed the influence of the threshold value for the co-occurrence weight used to measure the semantic relationship between two terms in the graph. Figure 1 shows the performance reached by our algorithm as a function of the significance threshold with and without the WordNet semantic enrichment.

The results shown in all experiments have been obtained using both the training set and evaluation scripts provided by Task 5 of SemEval-2010 (Kim, Medelyan, Kan, & Baldwin, 2010). The higher the threshold used, the smaller the number of relationships that occur among the terms. Therefore, the probability that there are isolated nodes or not connected to the graph is greater. Thus, isolated nodes are removed.

In addition, the smaller the threshold value, the greater the significance required by the co-occurrence of two words to be considered significant. Therefore, the method with smaller thresholds becomes more restrictive in selecting the representative words. This restricts the number of keyphrases than can be constructed from the selected words, improving precision at the expense of recall.

We can also see the evolution of the graph size as the threshold increases. As for the optimal threshold value, we can see that the highest F score is obtained with 2×10^{-3}. This value is used in the experiments hereafter. System performance increases because the significance threshold is higher. However, from the optimal point, the performance suffers a significant drop, probably because of the fact that some of the most significant terms of the graph begin to disappear.

Regarding the use of the WordNet semantic enrichment, we can see the improvement obtained in all cases, achieving an average improvement in F score of approximately 4%. Results indicate that WordNet provides a slight improvement,

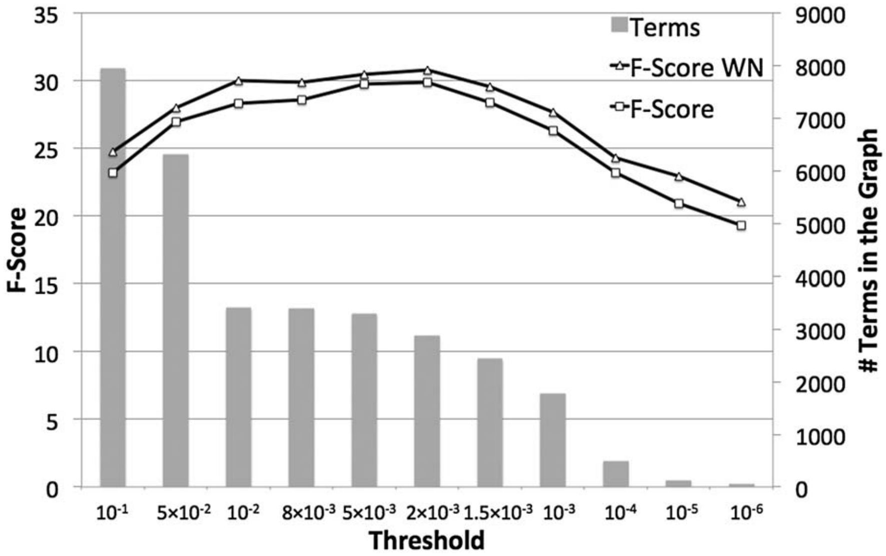

FIG. 1. Performance reached by our algorithm (line: F score; bars: graph size measured in number of terms) as a function of the significance threshold p with (F score WN) and without (F score) using the semantic enrichment.

but not too large. This could be because WordNet is biased toward generic words and would fail to capture proper nouns or technical terminology.

Extraction and Selection of Keyphrases

Most evaluation collections, including the one used here, are split into a training and a test part. Given that the method presented in this work is not supervised, we only use the training collection to adjust some parameters, such as threshold significance, with the aim to present the best performance of the system. However, these parameters could be extrapolated to other collections without the need to readjust them for optimal performance.

Candidate phrase preselection. Previous work (Lopez & Romary, 2010; Nguyen & Luong, 2010; Treeratpituk et al., 2010) has shown that some logical sections within a scientific article are more relevant than others when it comes to extracting keyphrases. Our method takes into account only the title, abstract, introduction, related work, future work, and conclusions. Once the parts of each document that will be analyzed have been identified, we proceed to extract candidate phrases. For this, our method does not use n-grams (contiguous sequence of n items from a given sequence of text), but extracts the longest sequence possible without overlapping. Accordingly, a word located in a particular part of a sentence can only belong to one keyphrase. A candidate phrase has to meet certain requirements. For the first term of the phrase:

- To be a noun or adjective
- To have a degree greater than 0 in the semantic relationship graph

The following terms are concatenated within the same phrase if they:

- Are a noun or adjective
- Are not a stopword, except "of," "for," or "to"
- Have a positive weight (link in the graph) between consecutive terms (non-stopword) higher than 0

Finally, a check is performed:

- If the phrase ends with "of," "for," or "to," the last term is removed.

When any of the above requirements is not fulfilled or the sentence ends because of a punctuation mark, the concatenation of the phrase finishes. This phrase becomes part of a list of preselected phrases and the selection process continues building a new phrase with the next term.

Figure 2 shows an example of candidate phrase preselection on a sentence extracted from an abstract from the evaluation collection. POS tags are displayed as a subscript after each term. The red underlined terms are the candidate keyphrases obtained. The numerical value above the terms corresponds to the weight between consecutive terms obtained from the semantic graph. In the case of the keyphrase composed of a single term, the degree is shown above the term.

Candidate phrase filtering. Although the definition of keyphrase does not imply maximum number of terms, it is not easy to find keyphrases with more than three or four terms in actual documents. Therefore, preselected phrases with more than three terms are discarded—without taking into account the stopwords "of," "for," and "to." Keyphrases extracted from the title, abstract, and introduction have a

DOI: 10.1002/asi

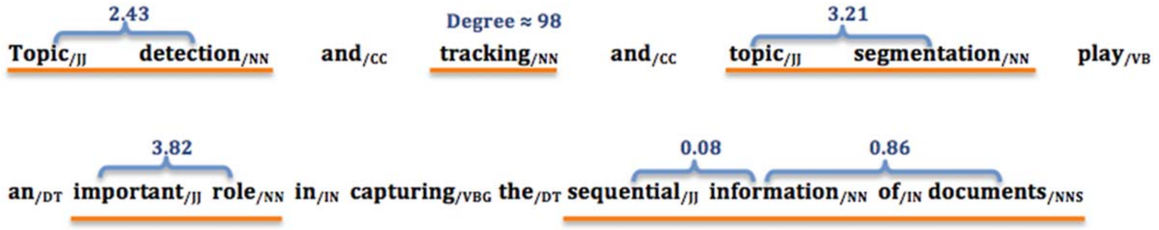
2.43
Degree ≈ 98
3.21
Topic/JJ detection/NN and/CC tracking/NN and/CC topic/JJ segmentation/NN play/VB
3.82
0.08
0.86
an/DT important/JJ role/NN in/IN capturing/VBG the/DT sequential/JJ information/NN of/IN documents/NNS

FIG. 2. Example of candidate phrase preselection. POS tags are displayed as a subscript after each term. The red underlined terms are the candidate keyphrases obtained. The numerical value above the term corresponds to the weight between consecutive terms obtained from the semantic graph or the degree in case of single terms. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

higher priority, giving special attention to those taken from the title. For this, we have added a push factor, adding a 10% to the weight of the keyphrases extracted from these parts of the document.

One of the main contributions of this work is precisely this stage of filtering. Although, in the next stage, a ranking is applied to the output of this filtering process, major performance improvements are obtained in this step.

To prove the effectiveness of the semantic relationship graph, we conducted a series of experiments that highlight its usefulness. The main goal is to select approximately one third of the preselected phrases by applying a cutting threshold to the values obtained from each method. We have therefore tested two term-weighting methods based on frequency and two using the semantic relationship graph:

- ***Tf.*** The terms in a keyphrase are weighed according to their frequency in the collection and then the average is obtained.
- ***Tf-Idf.*** Combines the definitions of term frequency and inverse document frequency to provide the average for every term.
- ***W-Link.*** From the semantic relationship graph, the average weight for the edges that link every pair of terms is obtained. In case there is no link between two terms, the default value is 0. Each term gets the average weight of its edges.
- ***PageRank.*** Assigns scores to each vertex in the graph according to the PageRank algorithm. To carry out the computation of PageRank, we used the JUNG[4]—Java Universal Network/Graph—framework. Parameter alpha (the probability of taking a random jump to an arbitrary vertex) was assigned to ".15" according to the typical values for alpha in documentation. Each keyphrase is weighted as the average of the PageRank values obtained from each term.

In the case of frequency-based methods (Tf and Tf-Idf), two collections were used for he experiments. We have used an English Wikipedia articles dump (May 2009)[5] as a reference collection. However, the collection for which best results were obtained was the one composed of the documents provided by the training data set.

In several articles (El-Beltagy & Rafea, 2010; Lopez & Romary, 2010; Nguyen & Luong, 2010; Treeratpituk et al., 2010) frequency-based methods and, more specifically, those based on Tf-Idf (Voorhees, 1993), exhibited the best performance. However, as seen in Table 1, through the use of our semantic relationship graph, significant improvements are obtained. The method based on PageRank achieves better results than the W-Link approach. The main drawback of the weighting link approach relative to PageRank is that not all the terms that occur together in a keyphrase have a link to each other in the graph.

TABLE 1. Performance measured in terms of precision (P), recall (R), and F measure (F) of the proposed filtering approaches over the top 15 candidates and the combined keywords provided by the SemEval-2010 Task 5.

Approach	P (%)	R (%)	F (%)
Tf	19.7	20.1	19.8
Tf-Idf	24.1	24.7	24.4
W-Link	29.4	30.1	29.8
PageRank	31.4	32.2	31.8

Keyphrase ranking. In the collection that we have used in this work, the number of keyphrases that are evaluated is predetermined. In this type of evaluation campaigns of keyphrase extraction systems, different analyses are usually made for the top 5, 10, and 15 proposed keyphrases. Thus, it is necessary to establish an order of priority among the selected keyphrases.

The method that we have followed for the ranking process is based on the frequency of occurrence of each keyphrase in the analyzed document. We calculate the frequency in the document of each keyphrase selected in the filtering process. Then, the keyphrases are ranked by frequency. Finally, the output of the algorithm is composed of the top *n* keyphrases, being *n* the maximum number of evaluated items depending on the collection used and the type of evaluation.

Sample output of SemGraph. Figure 3 shows a sample of the different phases for the extraction and selection of keyphrases from SemGraph. As can be seen, the first phase performs a preselection of candidate phrases and gets 91

[4] http://jung.sourceforge.net/

[5] http://download.wikimedia.org/enwiki/

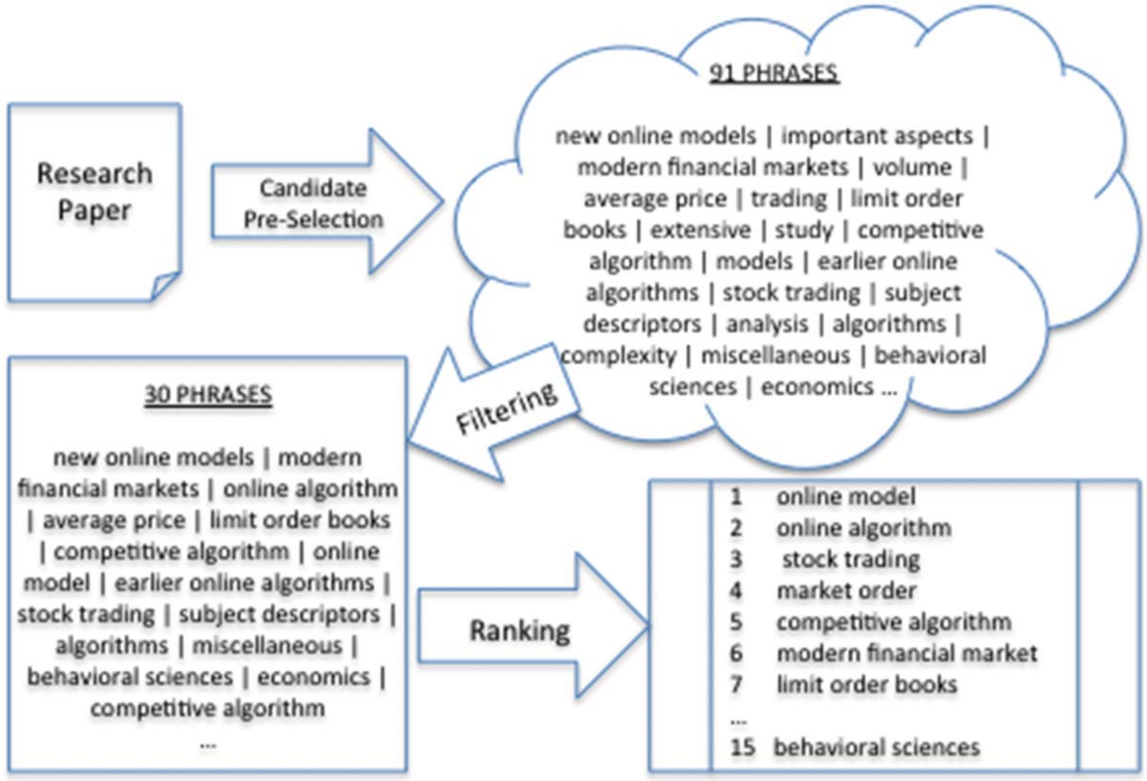
Research Paper
Candidate Pre-Selection
91 PHRASES
new online models | important aspects | modern financial markets | volume | average price | trading | limit order books | extensive | study | competitive algorithm | models | earlier online algorithms | stock trading | subject descriptors | analysis | algorithms | complexity | miscellaneous | behavioral sciences | economics ...
Filtering
30 PHRASES
new online models | modern financial markets | online algorithm | average price | limit order books | competitive algorithm | online model | earlier online algorithms | stock trading | subject descriptors | algorithms | miscellaneous | behavioral sciences | economics | competitive algorithm
...
Ranking
1 online model
2 online algorithm
3 stock trading
4 market order
5 competitive algorithm
6 modern financial market
7 limit order books
...
15 behavioral sciences

FIG. 3. Sample of the different phases for the extraction and selection of keyphrases from SemGraph on an article (Kakade et al. [2004]) in the collection used for the evaluation. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

TABLE 2. Sample of the output of SemGraph on a paper (Kakade et al. [2004]) in the collection used for the evaluation. Keyphrases extracted are compared with the reader-assigned keyphrases provided by organizers.

SemGraph output		
N	Detected keyphrases	Undetected keyphrases
1	Online model	Share
2	Online algorithm	Sequence of trade+trade sequence
3	Stock trading	Limit order book trading model
4	Market order	Volume-weighted average price trading model
5	Competitive algorithm	
6	Modern financial market	

items. Then, after filtering using PageRank, only 30 of the 91 items previously obtained remain. Finally, the ranking phase based on frequency generates 15 keyphrases that are the output of the system.

Next, we present a comparison of the output of the system with the reader-assigned keyphrases from the gold standard provided by organizers.

Table 2 shows a sample of the output of the system on an article (Kakade, Kearns, Mansour, & Ortiz, 2004) in the collection used for the evaluation. Keyphrases extracted by SemGraph are compared with the reader-assigned keyphrases provided by organizers. All reader-assigned keyphrases were extracted manually from the articles by annotators.

There are several reasons why the undetected keyphrases have been discarded by SemGraph. The first keyphrase ("share") is a single term and not very representative. The second and fourth keyphrases are composed of more than three terms and therefore are discarded. Finally, the third could be considered a keyphrase composed by three terms if the string "trade + trade" would be analyzed as a compound word and the stopword "of" would not be taken into account. However, the fact that it contains a symbol ("+") means that SemGraph does not consider this set of words as a keyphrase.

System Performance Analysis

One of the main contributions of this work is the way in which SemGraph is built. To evaluate the impact in the results of the method for constructing the graph, we conducted a set of experiments to compare the performance of our system with several graph-based algorithms designed for keyphrase extraction. As did Hasan and Ng (2010), we have reimplemented SingleRank (Wan & Xiao, 2008b) and ExpandRank (Wan & Xiao, 2008b) and a publicly available implementation of TextRank (Mihalcea & Tarau, 2004).

For every parameter, we used the same values that were used in the original works. For example, we used a window for TextRank from 2 to 10 words. Following Mihalcea and Tarau (2004) and Wan and Xiao (2008b), the best results were obtained with a co-occurrence window size for TextRank and SingleRank of 2 and 10, respectively. In the case of ExpandRank, the best performance was achieved by finding the 5 nearest neighbors for each document from the other documents in the corpus, thus following instructions provided by the researchers. The other parameters were set in the same way as in SingleRank.

Table 3 shows the results of the different graph-building algorithms used for the extraction of keyphrases in SemGraph. These results are based on a set of experiments in which we have used the same configuration as presented in earlier sections for the SemGraph system. The only difference between the experiments is the use of the corresponding algorithm for building the graphs required for the system. As can be seen, the four algorithms present competitive results. However, the use of the semantic-graph-based algorithm, which is presented in this work, shows a significant improvement over other algorithms.

TABLE 3. Performance in terms of F measure (F), of several graph-based algorithms over the top 15 candidates for the combined and reader-assigned keywords provided by the SemEval-2010 Task 5. Reader-assigned keyphrases were extracted manually from the articles. Combined keyphrases are composed by reader- and author-assigned keyphrases.

Graph-based algorithms used in SemGraph		
	F measure (SemEval-2010)	
Approach	Combined (%)	Reader-assigned (%)
TextRank	22.1	21.2
SingleRank	24.1	22.5
ExpandRank	25.6	22.8
Ours	31.8	30.5

We have also analyzed two other parts of our system. One comprises the sections of an article that our system considers as relevant. We conducted a set of experiments to test the impact on the results of these sections, as can be seen in Figure 4. This figure also shows the influence of the number of sentences extracted from each section. Regarding the sections used by the algorithm, our analysis starts by considering a basic package consisting of the title text, abstract, and introduction, because these sections are the main source of information for the extraction of keyphrases. From these sections, we have analyzed the influence of other sections such as the related work section, conclusions and future work, and methodology and experiments. As can be seen in the figure, the related work provides a slight improvement, somewhat higher than the conclusions. One of the most important aspects is that the use of these two sections together provides the most significant improvement, obtaining the best results. In contrast, the use of the combined section method and experiments slightly reduces system performance. The number of sentences taken from each section is also an important factor in system performance, and this is clearly shown in Figure 4. The system performance improves significantly as the number of sentences increases up to 10. However, beyond this point, the addition of new sentences worsens the performance of the system, perhaps as a result of the appearance of more-specific concepts in the text once the general concepts have been introduced in the first sentences.

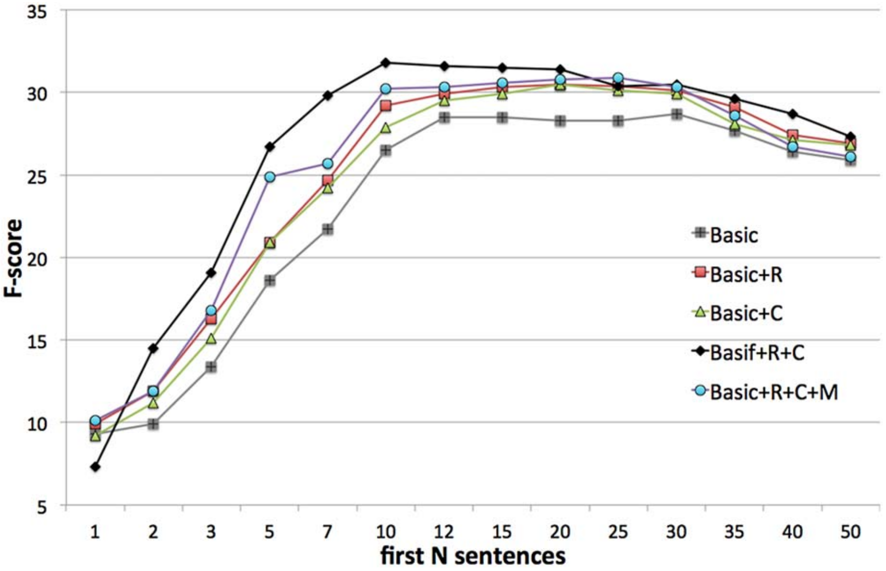

FIG. 4. Performance reached by our algorithm as a function of the sections taken into account and the number of sentences taken from each section in the document (Basic: title, abstract, and introduction; R: related work; C: conclusions and future work; M: methodology and experiments). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Experimental Results

Currently, there are several collections with which to evaluate keyphrase extraction systems: Inspec (Hulth, 2003), NUS (Nguyen & Kan, 2007), DUC2001 (Wan & Xiao, 2008a), and the ICSI meeting collection (Liu, Pennell, Liu, & Liu, 2009a). Despite the availability of several corpora, the scientific community has needed a standardized benchmark with well-defined training and test data sets with the main goal of maximize comparability of results.

Accordingly, the SemEval-2010 challenge Task 5 (Kim et al., 2010) compiled a set of 284 scientific articles from the ACM Digital Library with keyphrases selected by both their authors and readers. Data consisted of train, trial, and test data sets, composed of 40, 144, and 100 articles, respectively. Organizers collected data sets from the ACM Digital Library (conference and workshop papers). The input papers were selected from four different research areas and ranged from six to eight pages, including tables and images. Fifty annotators collected the actual reader-assigned keyphrases, and organizers assigned five articles to each annotator.

Three sets of "correct" keyphrases are provided for each article in each data set: reader-assigned keyphrases, author-provided keyphrases, and a combination of them. All reader-assigned keyphrases have been extracted manually from the articles, whereas some of the author-provided keyphrases may not occur in the content. Organizers evaluated the participating systems over the independently generated and held-out reader-assigned keyphrases, as well as the combined set of keyphrases (author- and reader-assigned). In this way, each run consisted of extracting a ranked list of 15 keyphrases from each document, ranked by their probability of being reader-assigned keyphrases.

In this section, we compare the performance of SemGraph with other works within the evaluation framework of SemEval-2010. Because there has been no subsequent use of the SemEval-2010 collection, and improved the results obtained by the participants in Task 5, we compare our system with the best works of that competition. Given that evaluation in this competition was carried out over the reader-assigned and combined (author- and reader-assigned) keyphrases, the performance of our system on these two sets is shown below.

Table 4 shows the results of our algorithm compared with the five systems that obtained the best performances on the combined set of keyphrases. Performance is measured in terms of precision, recall, and F measure, and the systems were ranked by F score. The best results over the combined and reader-assigned keyword sets were achieved by the HUMB system (Lopez & Romary, 2010). HUMB is a supervised system based on a tool for extracting bibliographical information from documents that also performs a content analysis based on phraseness, informativeness, and keywordness. Lexical and semantic features were extracted through a large-scale terminological database and Wikipedia.

TABLE 4. Performance measured in terms of precision (P), recall (R), and F measure (F) of the best five systems submitted to the SemEval-2010 Task 5 over the combined (author- and reader-assigned) keywords. Systems were ranked by F score over the top 15 candidates.

SemEval-2010: combined keywords			
System	P (%)	R (%)	F (%)
Humb	27.2	27.8	27.5
Wingnus	24.9	25.5	25.2
KP-Miner	24.9	25.5	25.2
Sztergak	24.8	25.4	25.1
ICL	24.6	25.2	24.9
SemGraph	32.4	33.2	32.8

TABLE 5. Performance measured in terms of precision (P), recall (R), and F measure (F) of the best five systems submitted to the SemEval-2010 Task 5 over the reader-assigned keywords. Systems were ranked by F score over the top 15 candidates.

SemEval-2010: reader-assigned keywords			
System	P (%)	R (%)	F (%)
Humb	21.2	26.4	23.5
Kx_Fbk	20.3	25.3	22.6
Sztergak	19.9	24.8	22.1
Wingnus	19.8	24.7	22.0
ICL	19.5	24.3	21.6
SemGraph	27.4	35.1	30.8

Wingnus (Nguyen & Luong, 2010) and Sztergak (Berend & Farkas, 2010) are also supervised systems with similar results that took advantage of the logical structure of documents and used external resources, such as Wikipedia, DBLP, and Google. The only unsupervised systems that appear in the top five are KP-Miner (El-Beltagy & Rafea, 2010) and KX (Pianta & Tonelli, 2010) that are based on several rules created from the observation, do not use external resources, and apply a frequency-based weighting approach (Tf-Idf and Idf, respectively).

SemGraph obtained an improvement of 5.3% on the F score with respect to HUMB. In addition, when compared with the best unsupervised system (KP-Miner) the improvement obtained is 7.6%. Analyzing the relationship between precision and recall, compared to the other systems, we can state that is balanced, discarding an improvement in one of these measures at the expense of the other.

Table 5 shows a comparison between our algorithm and the top five systems that obtained the best results on the reader-assigned set of keyphrases. Over this data set, HUMB is also the system that obtained the best results with an F score of 23.5%, our algorithm thus achieving an improvement of 7.3%. Regarding KX, which is the best unsupervised system, SemGraph improves 8.2% over its F score. Although not significantly, the improvement of our system with respect to HUMB corresponds mainly to Recall, obtaining a difference of 8.7%, compared to 6.2% in precision.

SemGraph does not use external resources outside the collection used for evaluation, therefore it can only propose keyphrases that appear in the documents of the collection. According to the organizers (Kim et al., 2010), the number of keyphrases from the combined set that did not appear in the text of the article is higher than in the reader-assigned set. The fact therefore that our algorithm improved more over the set of reader-assigned keyphrases with respect to the other systems suggests superior performance when our algorithm is used as a keyphrase extractor for the extractive summarization of text documents.

Conclusions and Future Work

In this article, we present SemGraph, an unsupervised keyphrase extraction algorithm. The main features of SemGraph are the use of a semantic relationship graph, constructed by considering each document as a coherent piece of information, and the absence of external resources for extracting keyphrases.

One of the main contributions of our system is its ability to identify words whose presence is statistically significant. The building of the semantic relationship graph is based on linking words appearing in the same document, provided their presence in the collection is statistically significant with respect to a null model. This semantic relationship graph allows us to carry out a more efficient term weighting than the frequency-based methods used up until now in most systems. Thus, a significant improvement is obtained in the filtering and ranking of keyphrases.

Most similar systems make use of external resources, such as Wikipedia, Google, and other terminological and linguistic resources. However, SemGraph uses only the documents in the evaluation data set and WordNet to enrich the relations between words in the graph with information from WordNet.

The result is a method that achieves an improvement of 5.3% and 7.3% over the two labeled sets of keyphrases used in the evaluation of SemEval-2010. In addition, if we compare SemGraph with the best unsupervised systems, an improvement of 7.6% and 8.2% is obtained, respectively.

In our future work, we intend to combine our algorithm with a method that proposes keyphrases that do not appear in the text of the analyzed documents. In addition, we will analyze the performance on other types of documents, such as news, Web pages, or tweets.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science and Innovation within the projects Holopedia (TIN2010-21128-C02-01) and EXTRECM (TIN2013-46616-C2-2-R) and the Universidad Nacional de Educación a Distancia (UNED) (2013-025-UNED-PROY).

References

Agirre, E., de Lacalle, O.L., & Soroa, A. (2013). Random walks for knowledge-based word sense disambiguation. Computational Linguistics, 40(1), 57–84.

Banerjee, S., & Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness. In IJCAI (Vol. 3, pp. 805–810). Acapulco, Mexico: Morgan Kaufmann.

Berend, G., & Farkas, R. (2010). Sztergak: Feature engineering for keyphrase extraction. In Proceedings of the Fifth International Workshop on Semantic Evaluation. SemEval '10 (pp. 186–189). Stroudsburg, PA: Association for Computational Linguistics.

Blei, D.M., Ng, A., & Jordan, M. (2003). Latent Dirichlet allocation. Journal of Machine Learning Research, 3, 993–1022.

Bougouin, A., Boudin, F., Daille, B. (2013). TopicRank: Graph-based topic ranking for keyphrase extraction. In International Joint Conference on Natural Language Processing (IJCNLP) (pp. 543–551).

DUC. 2002. Document understanding conference 2002. Retrieved from http://www-nlpir.nist.gov/projects/duc/

El-Beltagy, S.R., & Rafea, A. (2010). KP-Miner: Participation in SemEval-2. In Proceedings of the Fifth International Workshop on Semantic Evaluation. SemEval '10 (pp. 190–193). Stroudsburg, PA: Association for Computational Linguistics.

Fellbaum, C., 2010. WordNet: An electronic lexical database. Retrieved from http://www. cogsci. princeton.edu/wn

Gale, W.A., Church, K.W., & Yarowsky, D. (1992). One sense per discourse. In Proceedings of the Workshop on Speech and Natural Language. HLT '91 (pp. 233–237). Stroudsburg, PA: Association for Computational Linguistics.

Hasan, K.S., & Ng, V. (2010). Conundrums in unsupervised keyphrase extraction: Making sense of the state-of-the-art. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters. COLING '10 (pp. 365–373). Stroudsburg, PA: Association for Computational Linguistics.

Hirst, G., & St-Onge, D. (1997). Lexical chains as representations of context for the detection and correction of malapropisms. WordNet: An Electronic Lexical Database, 305, 305–332.

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of the 2003 Conference on Empirical Methods in NLP (pp. 216–223).

Jiang, J., & Conrath, D. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings of the International Conference on Research in Computational Linguistics (pp. 19–33).

Kakade, S.M., Kearns, M., Mansour, Y., & Ortiz, L.E. (2004). Competitive algorithms for VWAP and limit order trading. In Proceedings of the Fifth ACM Conference on Electronic Commerce (pp. 189–198). New York: ACM.

Kim, S.N., Medelyan, O., Kan, M.-Y., & Baldwin, T. (2010). SemEval-2010 task 5: Automatic keyphrase extraction from scientific articles. In Proceedings of the Fifth International Workshop on Semantic Evaluation. SemEval '10 (pp. 21–26). Stroudsburg, PA: Association for Computational Linguistics.

Leacock, C., & Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification. WordNet: An Electronic Lexical Database, 49(2), 265–283.

Lin, D. (1998). An information-theoretic definition of similarity. In ICML (Vol. 98, pp. 296–304).

Liu, F., Pennell, D., Liu, F., & Liu, Y. (2009a). Unsupervised approaches for automatic keyword extraction using meeting transcripts. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. NAACL '09 (pp. 620–628). Stroudsburg, PA: Association for Computational Linguistics.

Liu, Z., Li, P., Zheng, Y., & Sun, M. (2009b). Clustering to find exemplar terms for keyphrase extraction. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: EMNLP '09 (Vol. 1, pp. 257–266). Stroudsburg, PA: Association for Computational Linguistics.

Liu, Z., Huang, W., Zheng, Y., & Sun, M. (2010). Automatic keyphrase extraction via topic decomposition. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9–11 October 2010, MIT Stata Center, A meeting of SIGDAT, a Special Interest Group of the ACL. (pp. 366–376). Stroudsburg, PA: ACL.

Lopez, P., & Romary, L. (2010). HUMB: Automatic key term extraction from scientific articles in GROBID. In Proceedings of the Fifth International Workshop on Semantic Evaluation. SemEval '10 (pp. 248–251). Stroudsburg, PA: Association for Computational Linguistics.

Lopez, R.E., Barreda, D., Tejada, J., & Cuadros, E. (2011). MFSRank: An unsupervised method to extract keyphrases using semantic information. In I. Batyrshin & G. Sidorov (Eds.), Advances in artificial intelligence (Vol. 7094, pp. 338–344), Lecture Notes in Computer Science. Berlin/Heidelberg: Springer.

Martinez-Romo, J., Araujo, L., Borge-Holthoefer, J., Arenas, A., Capitán, J.A., & Cuesta, J.A. (2011). Disentangling categorical relationships through a graph of co-occurrences. Physical Review E, 84(4), 046108.

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. In Proceedings of EMNLP-04 and the 2004 Conference on Empirical Methods in Natural Language Processing. Barcelona, Spain: Association for Computational Linguistics.

Newman, D., Koilada, N., Lau, J.H., & Baldwin, T. (2012). Bayesian text segmentation for index term identification and keyphrase extraction. In Proceedings of COLING 2012 (pp. 2077–2092).

Nguyen, T.D., & Kan, M.-Y. (2007). Keyphrase extraction in scientific publications. In D.-L. Goh, T. Cao, I.T. Sažlvberg, & E. Rasmussen (Eds.), Asian Digital Libraries. Looking back 10 years and forging new frontiers (Vol. 4822, pp. 317–326), Lecture Notes in Computer Science. Berlin/Heidelberg: Springer.

Nguyen, T.D., & Luong, M.-T. (2010). WINGNUS: Keyphrase extraction utilizing document logical structure. In Proceedings of the Fifth International Workshop on Semantic Evaluation. SemEval '10 (pp. 166–169). Stroudsburg, PA: Association for Computational Linguistics.

Patwardhan, S., & Pedersen, T. (2006). Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In Proceedings of the EACL 2006 Workshop Making Sense of Sense-Bringing Computational Linguistics and Psycholinguistics Together (Vol. 1501, pp. 1–8).

Pianta, E., & Tonelli, S. (2010). KX: A flexible system for keyphrase eXtraction. In Proceedings of the Fifth International Workshop on Semantic Evaluation. SemEval '10 (pp. 170–173). Stroudsburg, PA: Association for Computational Linguistics.

Qazvinian, V., Radev, D.R., & Ozgur, A. (2010). Citation summarization through keyphrase extraction. In Proceedings of the 23rd International Conference on Computational Linguistics. COLING 2010 (pp. 895–903). Stroudsburg, PA: Association for Computational Linguistics.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In Proceedings of the 14th International Joint Conference on Artificial Intelligence (pp. 448–453).

Teufel, S., & Moens, M. (1998). Sentence extraction and rhetorical classification for flexible abstracts. In Spring AAAI Symposium on Intelligent Text Summarization (pp. 89–97).

Treeratpituk, P., Teregowda, P., Huang, J., & Giles, C.L. (2010). SEERLAB: A system for extracting key phrases from scholarly documents. In Proceedings of the Fifth International Workshop on Semantic Evaluation. SemEval '10 (pp. 182–185). Stroudsburg, PA: Association for Computational Linguistics.

Turney, P. (2000). Learning algorithms for keyphrase extraction. Information Retrieval, 2(4), 303–336.

Voorhees, E.M. (1993). Using WordNet to disambiguate word senses for text retrieval. In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 171–180). New York: ACM.

Wan, X., & Xiao, J. (2008a). CollabRank: Towards a collaborative approach to single-document keyphrase extraction. In Proceedings of the 22nd International Conference on Computational Linguistics COLING '08 (Vol. 1, pp. 969–976). Stroudsburg, PA: Association for Computational Linguistics.

Wan, X., & Xiao, J. (2008b). Single document keyphrase extraction using neighborhood knowledge. In Proceedings of the 23rd National Conference on Artificial Intelligence AAAI '08 (Vol. 2, pp. 855–860). AAAI Press.

Wang, X., Mu, D., & Fang, J. (2008). Improved automatic keyphrase extraction by using semantic information. In Proceedings of the 2008 International Conference on Intelligent Computation Technology and Automation ICICTA '08 (Vol. 01, pp. 1061–1065). Washington, DC: IEEE Computer Society.

Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., & Nevill-Manning, C.G. (1999). KEA: Practical automatic keyphrase extraction. In Proceedings of the Fourth ACM Conference on Digital Libraries (pp. 254–255). New York: ACM.

Wu, Z., & Palmer, M. (1994). Verbs semantics and lexical selection. In Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics (pp. 133–138). Stroudsburg, PA: Association for Computational Linguistics.