Exploiting extra-textual and linguistic information in keyphrase extraction

Gábor Berend
University of Szeged, Department of Informatics,
berendg@inf.u-szeged.hu

Submitted to Natural Language Engineering in 2014

Abstract

Keyphrases are the most important phrases of documents that make them suitable for improving natural language processing tasks, including information retrieval, document classification, document visualization, summarization and categorization. Here, we propose a supervised framework augmented by novel extra-textual information derived primarily from Wikipedia. Wikipedia is utilized in such an advantageous way that – unlike most other methods relying on Wikipedia – a full textual index of all the Wikipedia articles is not required by our approach, as we only exploit the category hierarchy and a list of multiword expressions derived from Wikipedia. This approach is not only less resource intensive, but also produces comparable or superior results compared to previous similar works. Our thorough evaluations also suggest that the proposed framework performs consistently well on multiple datasets, being competitive or even outperforming the results obtained by other state-of-the-art methods. Besides introducing features that incorporate extra-textual information, we also experimented with a novel way of representing features that are derived from the POS tagging of the keyphrase candidates.

1 Introduction

Keyphrases have the characteristic to describe and summarize the contents of documents in a compressed way. This makes them very appealing for several NLP tasks, including the categorization, summarization and retrieval of textual documents. Despite their potential utility, most of the documents are not supplied with keyphrases and their assignment to documents is time-consuming and costly, hence means for their automated generation are desirable.

Extracting keyphrases from documents has gained increasing academic interest in recent years. Although most of the previous studies focused on the domain of scientific papers, it is interesting to note that there has been studies dealing with the extraction of keyphrases from different genres of text – *e.g.* from news articles [45, 10, 7], meeting transcripts [19] and product reviews [2].

More formally, the task of keyphrase generation is to find a function k which determines a set of useful keyphrases K_i to document d_i, i.e. $k(d_i) = K_i$. Let set C_i consist of the candidate phrases (e.g. n-grams retrievable from a document up to a certain length) belonging to document d_i. Furthermore, let K_i^* be the set of gold standard keyphrases of document d_i. This set can be obtained from various sources, i.e. gold standard keyphrases might be regarded as those, which were assigned to a document by its authors or by some of its readers (e.g. in [16]). Gold standard keyphrases – although having the possibility of being less reliable – might even be derived from social tagging sites, such as CiteULike.org as it was done in [26].

As a final notation during the formal discussion of keyphrase generating techniques, let I be a set of index terms, the members of which are regarded *a priori* as phrases with the possibly to act as keyphrases on some document domain (e.g. scientific articles from the field of *game theory*). In the absence of any prior knowledge about the possible keyphrases, we can simply define a non-informative set of index terms by defining $I = \bigcup_{j \in N} \Sigma^j$, i.e. the infinite set consisting of all the possible character sequences of the alphabet Σ.

Imposing certain conditions on K_i – being the set of keyphrases returned for document d_i by mapping k – different approaches of automatic keyphrase generation can be distinguished:

- **Keyphrase assignment**: In this setting $K_i \subseteq \bigcup_{j \neq i} K_j^*$, meaning that the keyphrases assigned to a document are such ones, which are known to be gold standard keyphrases with respect some other document. Note that this approach does not require keyphrases returned for a document to be actually present in it, i.e. even $C_i \cap K_i = \emptyset$ might hold.

- **Keyphrase indexing**: In this setting $K_i \subseteq C_i \cap I$, meaning that the keyphrases proposed for document d_i should be present in it and be a member of some predefined list of index terms.

- **Keyphrase extraction**: In this setting $K_i \subseteq C_i$, the only difference being to keyphrase indexing that here the existence of some predefined list of index terms is not assumed (or equivalently a non-informative, infinite list of index terms is assumed).

In this paper, we present a feature rich keyphrase extraction framework for the scientific domain and provide an exhaustive evaluation on the Inspec [14] and the SemEval shared task data sets [16]. Our evaluations verified that besides benefiting from clues that are retrieved from the processed documents (*e.g.* by performing linguistic analysis), further improvements can be gained by the utilization of extra-document information derived from Wikipedia and WordNet. An on-line demo and the source code of our framework is made accessible from `http://rgai1.inf.u-szeged.hu/~berend`.

2 Related work

Here, we present the most common approaches used for keyphrase generation and also give an overview on previous attempts to exploit extra-document information in keyphrase extraction. Methods applied by top-ranked participants of the SemEval shared task on keyphrase extraction will also be introduced here for the purpose of comparison.

2.1 Keyphrase assignment

The automatic assignment of keyphrases (or tags) is often approached from a recommender system perspective. Prototypes of such solutions are AutoTag [29] and TagAssist [37]. The key idea behind approaches like these is to find similar documents and to assign tags of labeled documents to the unlabeled ones. AutoTag, one of the pioneering works of tag recommendation, applies standard information retrieval metrics to find similar documents and chooses tags from the nearest ones based on frequency information. As it turns out from [39, 9] many participants of the past ECML PKDD tag recommendation challenges also built their systems on document-similarity-based approaches.

These approaches thus assign keyphrases to documents that are not necessarily present in them, but which have been assigned to some similar documents previously. The reliance on reasonable amounts of similar documents makes such methods heavily domain-dependent, meaning that each and every time we would like to use them on a document set, tagged documents of the same genre are necessary.

2.2 Keyphrase indexing and extraction

Keyphrase indexing and extraction frameworks choose a set of candidate phrases within the documents then rank them in either a supervised or an unsupervised manner according to their keyphraseness and return the top ranked ones as the predicted keyphrases of a document. Keyphrase indexing strategies further require that the keyphrases suggested for a document should be contained in some domain-specific vocabulary or thesaurus.

Next, we present supervised and unsupervised approaches in more detail, as well as previous attempts to exploit extra-textual information for keyphrase extraction. Finally, we briefly describe some of the systems submitted to the SemEval shared task on keyphrase extraction, the training and test data of which was used during our evaluation.

2.2.1 Supervised and unsupervised solutions

GenEx [42] was one of the first systems to treat keyphrase extraction as a supervised learning task. It was a combination of the Genitor genetic algorithm and the module Extractor for extracting the keyphrases. Genitor was used in order to maximize the performance of Extractor by tuning the weights of 12

features that described keyphrase candidates. The work of [14] pointed out how linguistic knowledge can improve keyphrase extraction performance. One of her key findings was that the incorporation of the part-of-speech tags of the keyphrase candidates as features can result in a significant improvement in the quality of the extracted keyphrases. The model we propose here also relies on the POS tags of the keyphrase candidates, however, there are also major differences between the two approaches. Firstly, [14] used entire POS sequences as features, whereas here we split up POS sequences and generate multiple features out of them, based on their within-expression relative positions (for details, see Section 3.3.2). Yet another difference with respect how POS sequences were turned into feature values was that Hulth used the most frequent POS analysis of the candidates as features, while we treated it as a distribution over different analyses.

The statistics-driven approach in [40] used multiple language models to rank phrases based on their point-wise Kullback-Leibler divergence. This approach favored those phrases that received high probability values from a higher order in-domain (called the foreground) language model opposed to some unigram out-of-domain (called the background) language model, ensuring that highly ranked phrases were of sufficient *phraseness* and *informativeness*. This approach was intended to overcome the shortcomings of the *binomial log-likelihood ratio test* employed by [8] in order to find frequent collocations in text.

[28] introduces TextRank, which adapts the idea of the PageRank [33] algorithm for the extraction of important keyphrases and sentences from documents. This approach inspired many further researches, including [20, 4]. The authors of these papers introduced various unsupervised methods incorporating the simulation of random walks performed on the co-occurrence graphs built from keyphrase candidates. The unsupervised framework presented in [21] was also based on co-occurrences, however, its authors chose a clustering approach rather than a random walk-based one (we refer to this approach henceforth as KeyCluster).

The Topical PageRank (TPR) approach [20] first determines a set of latent topics based on some document collection, then handles documents as mixtures of those topics relying on Latent Dirichlet Allocation (LDA) topic modeling [3]. Then the topic-aware rankings of candidate terms are composed upon the determination of the keyphrases of a document with a certain topic distribution.

These approaches tend to perform well on the extraction of keyphrases from short text passages (*e.g.* from scientific abstracts), however, as reported in several previous works [13, 20, 4], their performances severely degrade when they are employed for the extraction of keyphrases from longer texts (*e.g.* full scientific papers). Our evaluation results presented in Section 4 suggest that our proposed solution has the advantage of performing consistently competitively to other state-of-the-art systems, irrespective of the document length from which keyphrases are extracted.

2.2.2 Solutions exploiting extra-textual information

Candidate phrase generation Existing systems can also be distinguished on the basis of the generation of candidate phrases. One way of carrying out candidate phrase extraction is the fully uncontrolled way, which means that basically any successive tokens – except for those starting or ending with stopwords or punctuation – are treated as candidate keyphrases, like it was done in KEA [47]. Different candidate phrase generating strategies are summarized in the paper of [15], which covers both aspects of candidate selection and feature engineering for the extraction of keyphrases from scientific articles.

Other systems may require phrase candidates to satisfy certain requirements – *e.g.* to be part of a noun phrase, as it was the case in [1]. The authors of [50] used the so-called core word expansion algorithm, which first finds a set of core words and the final set of candidate phrases are generated from these seed phrases. They claimed that their method might reduce the candidate set by about 75%.

The system KEA++ [27], however, uses a controlled indexing strategy, meaning that candidate phrases are retrieved with the help of a domain-dependent thesaurus. The use of a thesaurus can be thought as a way of incorporating extra-textual information into keyphrase extraction and its use prevents many ill-formed phrases from being handled as keyphrases. Having its advantages, this kind of approach may also exclude genuine keyphrases from the set of candidate terms and the availability of a topic-dependent thesauri is not necessarily the case for arbitrary domains. Domain dependent thesauri can be replaced by the use of Wikipedia as was done in [46], but despite its wide coverage, the possibility of the exclusion of proper keyphrases cannot be ruled out.

Handling semantic relatedness One of the key issues which have to be addressed in keyphrase extraction is that of recognizing semantic relatedness among candidate phrases and the documents in which they occur. The classic approaches are based on an analysis of term-document co-occurrence, involving *e.g.* Latent Semantic Indexing [18] or metrics derived from the path between the concepts of some taxonomy, usually from the hypernym tree of Wordnets, as in [35]. The articles of [34, 6] contain a detailed description of WordNet-related semantic relatedness measures. These approaches, however, might suffer from the lack of desirable coverage of the taxonomies that they employ.

Extra-textual information was also used in [43] via the employment of web queries in order to increase the consistency of the phrases extracted from documents. Other works, including [24], incorporated information derived from citations into their models to improve their results.

Wikipedia was widely used earlier in tasks that attempted to determine semantic relatedness among concepts, *e.g.* [38, 12, 49, 30]. Our proposed approach is related to these earlier studies by the fact that we employ the category hierarchy of Wikipedia concepts as an external source of information to improve the quality of our keyphrase extraction framework.

The work presented here belongs to those supervised keyphrase extraction

approaches, which use various ways of controlling the candidate phrase set, but it is not done by means of any thesaurus. Semantic knowledge is also incorporated into our system by defining features based on the category hierarchy of Wikipedia and normalizing the keyphrase candidates relying on the synsets of WordNet.

2.2.3 Systems participating the SemEval shared task on keyphrase extraction

A complete description of the participating systems can be found in the overview paper written by the shared task organizers [17]. To the best of our knowledge, the results achieved by the best-performing shared task participants can still be regarded as state-of-the-art and more recent attempts – including [48, 4] – have not been successful in surpassing them. For this reason, we will present the results of top-performing shared task participants for comparative purposes in Section 4.2.1.

Maui [26] arguably lies the closest to our approach as it retrieves various metrics from Wikipedia to use them as features describing the keyphraseness of candidate phrases. The clear distinction between Maui and our approach is that while Maui uses all the textual content of Wikipedia (*e.g.* by calculating the probability of finding some keyphrase candidate as an anchor text), we only rely on its category hierarchy. The approach applied by Maui needs a massive index of textual occurrences from a Wikipedia dump – something that our approach does not require, still being able to outperform it.

Some of the highly ranked shared task participants, like HUMB [22] and WINGNUS [32] crawled the original PDF articles and processed them – instead of relying on the plain text versions provided by the organizers – which gave them the chance to examine the logical structure of documents more precisely. Despite the fact that our approach did not enjoy such benefits, it performed competitively or even better than these systems and a possible line of future research might be to focus on the combination of semantical and structural features derived from documents.

HUMB not only used Wikipedia as an external resource, but also GRISP [23] – being a large-scale terminological database derived from multiple resources – as a mean for discriminating between proper and improper keyphrases. In that work some of the Wikipedia-based features, that were originally introduced in Maui, were employed as well.

3 Keyphrase Extraction Framework

In our study, the supervised machine learning approach for the extraction of keyphrases was employed. Candidate terms were extracted from the articles and those present among the set of gold annotation keyphrases were treated as positive training examples. The sets of gold annotation keyphrases were included in the datasets we used for training and evaluation (see Sections 4.1.1

and 4.1.2 for details).

Maximum Entropy classifiers were then trained where we set the class labels of the keyphrase candidates according to the set of gold annotation keyphrases. Finally, the top-n keyphrase candidates with the highest probability values of belonging to the class of proper keyphrases were the predicted keyphrases for a given test document. We used the machine learning framework of MALLET [25] to train our models, which distinguish proper keyphrases from improper ones.

Although Naïve Bayes models are frequently employed for keyphrase extraction (see *e.g.*[47, 27, 26]), we primarily used conditional modeling here in the form of a Maximum Entropy classifier, as some parts of the feature set incorporated in our framework could easily violate the conditional independence assumption of Naïve Bayes models. Below, we will describe how keyphrase candidates and the feature space representing them were constructed.

3.1 Candidate term generation

One key aspect in keyphrase extraction is the way keyphrase candidates are selected and represented. As a high imbalance usually exists among the number of potentially extracted n-grams and the actual number of genuine keyphrases for some text, keyphrase candidates should be filtered instead of using any successive n-grams.

In our definition, keyphrase candidates were the n-grams that were not longer than 5 tokens and started and ended with a non-stopword token having one of the POS tags of noun, adjective or verb. All the other phrases besides the start and end tokens either had to be present on a list of stopwords (containing elements of closed word classes such as prepositions and determiners) or tagged as either noun, adjective or verb. Some phrases that fulfilled the above-mentioned criteria were still discarded due to the positional rule that phrases only present in the *References* part of an article were not treated as keyphrase candidates.

Once we had the keyphrase candidates, they had to be converted into their normalized forms. The normalization of an n-gram consisted of lowercasing and Porter-stemming each of the lemmatised forms of its tokens, then putting these stems into alphabetical order (while omitting the stems of stopword tokens). With this kind of representation, it was then possible to handle two orthographically different, but semantically equivalent phrases, such as *diffusion of innovation* and *Innovation diffusion* in the same way, i.e. in the normalized form of *innov diffus*. For the linguistic analysis of the articles (i.e. tokenization, lemmatization, POS tagging) we used the Stanford CoreNLP API [41].

As stated above, all the sequences of tokens consisting of the allowed POS tags or stopwords (not at the beginning or the end of a token sequence) were regarded as candidates. The reasons why we did not restricted candidates to be of some more precisely defined, limited set of linguistic patterns, *e.g.* to solely regard candidates which matched certain POS patterns (such as `(NN(S)?|JJ)? NN(S)?`) were the following:

- We noticed that proper keyphrases were sometimes erroneously tagged with POS sequences that would not match any POS patterns typically employed *e.g.* in [14]. For example, a phrase like '*simulated annealing*' might be tagged as '*VBN VBG*' (as opposed to '*JJ NN*'). Since there were only a few gold standard keyphrases assigned to each document, ruling out proper keyphrases from the set of candidates could severely decrease our recall values. For this reason, we rather favored the kind of permissive candidate generation strategy described above, in order to avoid the elimination of proper keyphrases at the cost of generating more improper candidates.
- Even if there was a sequence of tokens that was assigned the correct POS tags, we could still benefit from token sequences that were tagged with POS sequences that are otherwise untypical of proper keyphrases. Observing the sequence '*making decisions*' (assigned with the proper POS tags '*VBG NNS*') in some text could be used to update the feature statistics of the sequence '*decision making*' (being a proper keyphrase of POS tags '*NN NN*') as the normalized versions of both phrases collide to the form '*decis make*'.

3.2 Filtration of the candidate set

As mentioned previously, the number of potentially extracted keyphrase candidates might exceed the number of proper keyphrases by orders of magnitude for a document. Treating keyphrase extraction as a supervised learning task and not being circumspect on the candidate phrase generation might result in the fact that the more interesting class of proper keyphrases might be easily underrepresented, which might affect the performance of their identification negatively. A possible way to overcome this problem is to restrict the extraction of candidate phrases, i.e. filter them in such a way that as many genuine keyphrases are turned into classification instances as possible, while ruling out as many improper sequences of words as possible. The methods listed here contain restrictions based on stopwords and the utilization of WordNet to incorporate semantic knowledge. The above mentioned candidate phrase restricting policies are to be presented next in more details.

3.2.1 Introducing stopword rules

Our initial definition of candidate phrases, (i.e. those n-grams which both start and end with either a *noun*, *adjective* or *verb*) did not say anything about the tokens with indices $n - 1 \geq i \geq 2$ for n-grams of $5 \geq n \geq 3$. This restriction dealt with the elimination of keyphrase candidates that were highly unlikely to serve as proper keyphrases based on the relative frequency of how often they contained a stopword.

The assumption here was that proper keyphrases occur at least once within an article without including a stopword. So, for instance, the previously mentioned normalized form, *innov diffus* would be discarded for a document if all

of its occurrences were in the form of *e.g. diffusion of innovation* (i.e. containing the stopword *of* in all of its occurrences). However, if there was one single occurrence of *innovation diffusion*, then its normalized form was not excluded from becoming a phrase candidate. With the help of this filtering step, normalized n-grams belonging to the class of improper keyphrases, such as *basi method* – being the normalization of the n-gram *basis of the method* – could be excluded from the list of keyphrase candidates.

3.2.2 Incorporating WordNet knowledge

Experiments relying on the usage of WordNet [11] were also conducted in order to provide an extended way of normalizing phrases. In these settings, the normalized form of a single token was determined by first searching for all its synsets (in the case of verbs, these were such noun synsets that were in derivative relation with the synsets of the verbal word form). Then, instead of Porter-stemming the lemma of an original token, its most frequent word form was stemmed. The most frequent word forms were determined based on the estimated frequencies of WordNet for all the word forms among the synsets belonging to the original token (or its noun derivative synsets in case of verbs). In this way, two – originally differently stemmed – word forms, such as *optimize* and *optimum* could be stemmed to the same root forms. Another advantage of this procedure is that it is able to handle semantic similarity to some extent due to the fact that a word form is treated as if it were the most frequent word form among its synsets (*e.g.* the word form *task* is treated as if it were the word form *job*).

3.3 Description of the feature space

When designing our system, we employed a supervised learning approach for keyphrase extraction, where the keyphrases of documents are determined by first identifying sets of candidate phrases, then classifying their elements as either proper or improper keyphrases, based on the prediction of a machine-learned model.

To provide a baseline to our solution, we implemented the basic feature set of KEA [47] as it is one of the most cited publicly available tool for supervised keyphrase extraction. We did not use the KEA framework itself as we employed a different strategy for generating keyphrase candidates, but rather reimplemented its basic features in our system. These features are the tf-idf score and relative first occurrence (i.e. the quotient of the first token position of a keyphrase candidate and the length of the whole document – expressed in tokens – which contains it).

Our baseline solution also incorporated the use of the standard deviation of the start token positions of keyphrase candidates, which is also an optional feature in the KEA framework. This feature takes on smaller values if a keyphrase candidate is mentioned only at some well-bounded region of a document and takes higher values when a keyphrase candidate is mentioned repeatedly at var-

ious points of a document. Phrases that are more important and might serve as keyphrases tend to be used repeatedly, *e.g.* in the introduction and conclusions as well.

3.3.1 Wikipedia-derived features

Wikipedia provides a deep insight into human knowledge, which suggests that it could be used in the determination of keyphrases of scientific documents.

Utilizing Wikipedia categories As a first attempt, candidate phrases had a binary feature to indicate whether there existed an article on Wikipedia of the same title. If a Wikipedia article could be assigned to a classification instance, it was intended to suggest that it was representing such a general and well-known concept that it might be worth applying this concept as a keyphrase.

However, this approach could not improve the system performance, therefore, other ways of utilizing Wikipedia were experimented with. The reason why the previous feature was unable to result in improvement in the classification of keyphrases might be due to the highly detailed nature of Wikipedia, i.e. it also has articles for such common concepts as `results` or `studies`, which phrases are frequently used in scientific literature, but rarely function as proper keyphrases.

A more sophisticated way of exploiting Wikipedia involved the use of its categories. (Wikipedia categories form a taxonomy, indicating which article belongs to which (sub)category). Instead of simply representing it as a binary feature that (at least) one Wikipedia article could be assigned to a candidate phrase, all the nominal parts of the normalized titles of Wikipedia categories for its related Wikipedia articles were added as separate binary features to the feature space.

The normalization of the Wikipedia category names was similar to that of keyphrase candidates (see Section 3.1). Table 1 contains features induced for a likely and an unlikely normalized n-gram form. We observed that approximately 37% of the keyphrase candidates which belonged to the class of proper keyphrases could be assigned to at least one Wikipedia category, whereas the same proportion for improper keyphrase candidates was 13%. Likely and unlikely features derived from the category structure of Wikipedia were not explicitly distinguished within the feature set, but were expected to be assigned high absolute-valued feature weights during the training phase, based on their co-occurrence of the proper and improper keyphrase aspirants. Throughout our experiments we refer to the above-described features with the name, *WikiCategory*.

Utilizing multiword expressions (MWEs) from Wikipedia Multiword expressions are lexical items that can be decomposed into single words and display idiosyncratic features [36], in other words, they are lexical items that contain spaces. The fact that multiword expressions often turn out to be proper keyphrases implies that the knowledge of MWEs in a given text can be exploited

Table 1: Example features induced based on the category hierarchy of Wikipedia

Normalized candidate	Wikipedia article	Example Wikipedia categories
distribut hash tabl	Distributed hash table	Distributed data-storage File sharing
result	Results	1989 albums Pet Shop Boys albums Epic Records albums

in the determination of keyphrases. However, we should add that the two tasks (i.e. finding the MWEs and the keyphrases of documents) should be treated differently, since not all multiword expressions necessarily behave as keyphrases in every context (*e.g.* although the phrase *research group* is definitely an MWE, its use as a keyphrase when it is present in the affiliations part of a scientific paper is not likely to act as a proper keyphrase for a document).

To be able to decide which phrases might function as MWEs, a wide list of possible MWEs were collected from Wikipedia: all the formatted (i.e. bold or italic) and anchor texts of links from Wikipedia that was at least two tokens in length, starting with lowercase letters and contained only English characters or some punctuation, were collected.

Having constructed that list, an alignment of its elements and the corpus was carried out (handling linguistic alternations as well), regarding those n-grams as genuine MWEs that started and ended with tokens of either a noun or adjective POS tag and had no other (possibly zero) tokens in between them that were tagged as either noun, adjective, preposition or possessive ending.

To demonstrate the added value of MWEs in the task of keyphrase extraction, binary features were introduced to indicate whether a certain n-gram (1) was an MWE, (2) could be built up from more MWEs, or just simply (3) was the superstring of at least one MWE from the list. Hence, when deciding on the MWE-related features of a keyphrase candidate, we need to decide whether it

- is annotated by the automatic process as an MWE in its full length (based on the MWE list extracted from Wikipedia and the POS sequence of a candidate phrase, *e.g. maximal social welfare ratio*),
- can be assembled from two MWEs of the list (*e.g. resource allocation problems*, where *resource allocation* and *allocation problems* were in the list separately, but not as one phrase),
- can be a superstring of at least one MWE (*e.g. general analysis remains*, due to the presence of *general analysis* on the list of MWEs).

Throughout our evaluations, we refer to this set of features with the name, *MWE*. During our experiments, we found that 34% and 9.6% of the keyphrase candidates belonging to the class of proper and improper classes, respectively, were assigned with any of the MWE-related features.

3.3.2 Linguistic and orthographic features

As some POS tags are more frequent than others within the class of proper keyphrases, the authors of [14, 31] also proposed to derive features from the POS tags of keyphrase candidates. Features generated by the POS tags belonging to the tokens of different orthographic occurrences of a normalized phrase were applied in our study as well. Entire POS tag sequences seem to be more informative compared to the simple indication of the presence of POS tags in an n-gram, but it is also true that taking all the combinations of POS sequences up to a certain length as separate features might invoke data sparsity issues.

To overcome this problem, POS tagging-derived features incorporated the positional information of tokens as well. Features of POS tags that were assigned to a token being itself a 1-token long keyphrase candidate, at the beginning, at the end and inside an n-gram, got a prefix of *S-*, *B-*, *E-* and *I-*, respectively. For instance, the phrase *dynamic/JJ semantics/NN* induces the features *B-JJ*, *E-NN* to fire, whereas the 1-token-long phrase *semantics/NN* induces the feature *S-NN* to do so. This way, POS features were expected to contain probably less information, but to behave better with respect to dimensionality. In order to see the differences between the two approaches, both sequential and non-sequential POS tagging feature representations were implemented and evaluated within the framework.

A set of binary features were implemented that was related to the orthography and semantics of keyphrase candidates, as Named Entities (NEs) usually both have special orthographic characteristics and special semantic roles in their content. The position of NEs within candidate phrases was encoded in these features in a similar way as it was achieved for POS tags: separate features were created to indicate whether an n-gram contained a certain type of NE-class located at the beginning (B), inside (I) or at the end (E) of a keyphrase candidate. A special symbol for single token (S) keyphrases candidates was also reserved. For instance, the phrase *Nash* had the feature *S-PER* set to true, while *Nash equilibrium* had the feature *B-PER* set as true (and *S-PER* as false, naturally).

Proper keyphrases often have other special orthographic characteristics, *e.g.* it is the case with *UDDI* (being an acronym of the technical term *Universal Description Discovery and Integration*). Owing to the fact that not just the normalized but the original forms of the candidate phrases were stored in our representation, it was possible to construct two features for this: the first feature was responsible for character runs (i.e. more than 2 of the same consecutive characters), and another is responsible for 'strange capitalization' (i.e. the presence of uppercase characters besides the initial one). The *I-*, *O-*, *B-*, *S-* prefixes were applied here as well, just like for the Named Entity-related features. Together with the NE-related features, these features formed the ones that we refer to as *Orthography* features hereinafter.

4 Experimental results and discussion

Throughout our evaluations, we experimented with features based on

- the Wikipedia categories assigned to keyphrase candidates (referred to as *WikiCategory* features)
- the relation of keyphrase candidates to a list of Wikipedia-derived multiword expressions (referred to as *MWE* features)
- the part-of-speech sequences keyphrase candidates were analyzed (referred to as *POS* features)
- the Named Entity and surface form characteristics of keyphrase candidates (referred to as *Orthography* features).

4.1 Datasets

In order to conduct thorough experiments, we used two benchmark datasets on keyphrase extraction from scientific documents. In the followings, we introduce these datasets in more details, then we report our experimental results obtained using them.

4.1.1 SemEval shared task dataset

The primary dataset we used to test the effectiveness of our approach was the dataset of the SemEval-2 shared task on keyphrase extraction [16]. This dataset is a subset of the ACM Digital Library and consists of 244 scientific papers of length ranging from 6 to 8 pages taken from four different research areas (i.e. Distributed Systems, Information Search and Retrieval, Distributed Artificial Intelligence – Multiagent Systems, Social and Behavioral Sciences – Economics).

The set of documents was split into a training set of 144 documents and a test set of 100 documents by the organizers of the shared task. Sets of gold standard keyphrases assigned by both the original authors and undergraduate CS student readers of the publications were included in the dataset, which made supervised training possible.

Although the scope of the shared task was keyphrase extraction, there were certain elements in the gold standard set of keyphrases that were not present in the test documents. The organizers of the shared task report in their task description paper [16] that 19% of the gold standard keyphrases did not actually appear in the documents, implying that keyphrase extraction techniques could not achieve a recall score more than 0.81. [16] also reports that the reader-defined sets of keyphrases achieved a precision and recall score of 0.215 and 0.778, respectively, when compared to the sets of keyphrases assigned to the publications by their authors. The previous scores – that were achieved as a result of a human tagging activity – yield an F-score of 0.336.

As the primary ranking criterion at the shared task was based on the evaluation against the reader-assigned keyphrases, we regarded those phrases as the gold standard set of keyphrases during training phase. Further evaluations when the gold standard keyphrases were identified as the union of the author and reader-assigned phrases of the documents were also carried out. We shall refer to the latter type of gold standard annotation as the combined one. We shall also note that there was often a substantial overlap between author and reader-assigned keyphrases and that the amount of author and reader-assigned keyphrases differed substantially (i.e. their average numbers were 4 and 12, respectively).

4.1.2 Inspec dataset

The other keyphrase extraction dataset that we used for the evaluation of our approach is a subset of the Inspec database. It was originally created for the experiments of [14] and it consists of 2,000 scientific abstracts with both controlled and uncontrolled sets of keyphrases determined by professional indexers. The elements of the controlled set of keyphrases are required to be present in a thesaurus of index terms, whereas uncontrolled keyphrases were terms freely assigned to articles by the indexers.

The document collection is split into a training set of 1,000 abstracts and development and test sets consisting of 500 abstracts each. As we wanted to see the general applicability of our proposed model – that was primarily intended to perform well on the SemEval dataset – we simply discarded the development set and trained a model with the very same settings as we did for the SemEval dataset. Following the evaluation strategy most often employed in previous researches including [14, 28, 21, 20], we also used the uncontrolled keyphrases for evaluation purposes (as only 18% of the controlled keyphrases are present in the abstracts as opposed to more than 76% for the uncontrolled terms).

4.2 Experiments

As [13] also points it out, different authors performing their evaluation on the Inspec dataset calculated the recall value of their systems differently, which makes the direct comparison of their performances difficult. The different ways of calculating the recall score of a system are the following:

- Permissive evaluation – employed in [14, 21] for instance – requires only those gold standard phrases to be predicted by a system to achieve a perfect recall that can be found within the abstracts.
- Restrictive evaluation – employed in [28, 13] for instance – does not take into consideration whether the gold standard keyphrases can be found in the abstracts; in the case a gold standard keyphrase is not returned by a system, it is counted as a false negative decision under any circumstances.

In most of the cases, it is clear what kind of evaluation was employed by the authors of previous works, as it is either stated explicitly, or it can be inferred

from the results. There are some unfortunate cases, however, when the criterion under which authors report their results is not entirely clear.

For the above mentioned reasons, we regard our results based on the official evaluation script and relying on the standard benchmark dataset of the SemEval shared task more suitable for the comparison of the performances of different approaches. The other reason why we regard performing comparisons on the SemEval dataset more favorable is that it contains full documents as opposed to the Inspec dataset, which consists of scientific abstracts. This can be important, as several previous studies suggested [13, 20, 4] that systems performing well on the extraction of keyphrases from short documents, often suffer a substantial loss of performance when they need to extract keyphrases from long documents. Nevertheless, results on the Inspec dataset might provide interesting additional insights to the performance of our framework.

We should also add, that although the appropriateness of both the permissive and the restrictive evaluations can be argued, we consider the latter kind of evaluation to be more appropriate, as this way only those systems can be awarded with a perfect recall that return all the keyphrases determined by a professional indexer (irrespective of the fact whether the gold standard keyphrases are present in the documents). For this reason, we report our results using the restrictive evaluation schema and in Table 8, we also explicitly indicate the kind of evaluation that was employed in other papers.

4.2.1 Evaluation on the SemEval dataset

We conducted evaluations in the exact same manner for the SemEval dataset as they were performed at the shared task by using the evaluation script provided by the organizers. This kind of evaluation measured the precision, recall and F-score values of the stemmed forms of the top-n ($n \in \{5, 10, 15\}$) ranked keyphrases. We regarded those keyphrase candidates as the top-n ones that were assigned the top-n highest probability values of belonging to the class of proper keyphrases by our log-linear classifier.

We built our baseline model based on the feature set of KEA and added one feature at a time, to learn their contribution to the performance. The effects of extending the baseline feature set with one of the features described in Section 3.3 are illustrated in Table 2 and 3 for the evaluation on the SemEval dataset against the reader-assigned and combined gold standard keyphrases, respectively.

Our models that use one additional feature besides the ones also included in the baseline approach consistently beat the performance of our baseline system with a large margin for all the evaluation scenarios. Nevertheless all of the proposed features proved their usefulness, it was important to see their combined effect towards the performance.

The results of our classifier can be seen in the first lines of Table 4 and 5 when combining all the features into a single model and evaluating it on the SemEval dataset against reader-assigned and combined gold standard keyphrases, respectively. In these tables *Merged* refers to the fact that these models

Table 2: Results obtained by adding one extra feature to our baseline feature set at a time, evaluated against reader-assigned keyphrases of the SemEval dataset

	Top-5			Top-10			Top-15		
Method	P	R	F	P	R	F	P	R	F
Baseline	14.8	6.2	8.7	10.0	8.3	9.1	8.2	10.2	9.1
Baseline+WikiCategory	23.2	9.6	13.6	18.5	15.4	16.8	15.7	19.6	17.5
Baseline+MWE	18.0	7.5	10.6	14.3	11.9	13.0	10.9	13.5	12.1
Baseline+POS	26.6	11.1	15.6	22.7	18.9	20.6	18.8	23.4	20.9
Baseline+Orthography	28.0	11.6	16.4	21.3	17.7	19.3	16.9	21.1	18.8

Table 3: Results obtained by adding one extra feature to our baseline feature set at a time, evaluated against combined keyphrases of the SemEval dataset

	Top-5			Top-10			Top-15		
Method	P	R	F	P	R	F	P	R	F
Baseline	19.2	6.6	9.8	13.4	9.1	10.9	10.9	11.1	11.0
Baseline+WikiCategory	31.4	10.7	16.0	24.4	16.6	19.8	20.4	20.9	20.6
Baseline+MWE	23.4	8.0	11.9	17.7	12.1	14.4	13.4	13.7	13.6
Baseline+POS	33.2	11.3	16.9	27.8	19.0	22.5	23.1	23.6	23.3
Baseline+Orthography	35.6	12.1	18.1	26.5	18.1	21.5	21.3	21.8	21.6

merged all the previously described features into a single model. Examining these results, we can see that the gains of the different views of the candidates were able to add up and yield a yet improved performance.

Merging all the feature templates together resulted in a feature set that consisted of more than 30,000 elements. One of the reasons behind this was the use of entire POS and named entity tag sequences as features and the other was the usage of the Wikipedia categories. Feature counts on that (and even much bigger) scale are not irregular in natural language processing tasks, but if we add that the training set contained approximately 2,000 instances belonging to the class of proper keyphrases, the need for the reduction of the number of features can be argued.

In order to empirically test our hypothesis about data sparsity when using a rich feature set, we replaced features which encoded entire sequences of tags with a series of per token position-label pairs, as described in Section 3.3.2. The second rows (marked with the *BIES* subscript) in Table 4 and 5 list the results obtained when non-sequential tag features were applied instead of the sequential ones. Using non-sequential features not only reduced the dimensionality of the feature space, but also slightly improved the quality of the keyphrases which were returned as the best 15 phrases. As the main ranking criterion of the systems participating at the shared task was based on the performance of their top-15-ranked keyphrases, this kind of feature representation was employed in our subsequent experiments.

Next, the effects of the candidate filtration (*CF* for short) techniques, as described in Section 3.2, were also examined. Candidate filtration lessened the effect of the highly dominant nature of the non-proper training instances. As a

Table 4: Effect of the use of the non-sequential features and candidate selection against the reader-assigned gold annotation on the SemEval dataset

	Top-5			Top-10			Top-15		
Method	P	R	F	P	R	F	P	R	F
Merged	31.6	13.1	18.5	24.5	20.4	22.2	19.5	24.3	21.6
Merged$_{BIES}$	31.2	13.0	18.3	23.9	19.9	21.7	19.9	24.8	22.0
Merged$_{BIES+CF}$	32.4	13.5	19.0	23.8	19.8	21.6	20.0	24.9	22.2

Table 5: Effect of the use of the non-sequential features and candidate selection against the combined gold annotation on the SemEval dataset

	Top-5			Top-10			Top-15		
Method	P	R	F	P	R	F	P	R	F
Merged	39.0	13.3	19.8	30.4	20.7	24.7	24.4	25.0	24.7
Merged$_{BIES}$	39.2	13.4	19.9	30.2	20.6	24.5	24.9	25.4	25.2
Merged$_{BIES+CF}$	40.6	13.9	20.7	30.2	20.6	24.5	24.9	25.4	25.2

result of applying the proposed techniques, over 45% of the training instances were discarded (see Table 6), but the quality of the extracted keyphrases remained at the same level or even increased, as it can be seen in the third rows of Table 4 and 5.

As regards comparative results to the performance of shared task participants, our system performed as well as any of them when evaluated at the level of top-5 keyphrases, and it was only the system HUMB [22] – which used extra training data besides the corpus provided by the organizers – that achieved better performances against all the three (i.e. reader, combined and author) gold standard sets at the level of top-10 keyphrases (see Table 7). Our system ranks second – again behind HUMB – for evaluations against the top-15 keyphrases. The paper describing HUMB reports that their combined test set performance evaluated for the top-15 keyphrases improved by 7.4% due to the additional training data they used.

Looking at the results of WINGNUS and Maui systems, it is interesting to note that WINGNUS tends to perform better on evaluations against the reader-assigned keyphrases, while its relative performance degrades severely on evaluations against author keyphrases and the opposite holds for Maui. The performance of our system, however, seems to exhibit a more robust performance over different evaluation settings.

The official ranking of the shared task was based on the top-15-ranked keyphrases. However, as both the median and the mode of the number of gold standard keyphrases on the test set were below 15 – i.e. they were 12 and 4 for reader and author-assigned keyphrases, respectively – we think that evaluations performed at some lower threshold are more relevant when judging the utility of keyphrase extraction systems on this dataset.

The following example demonstrates the strictness of the evaluation applied in the shared task for one of the test set documents – entitled *Trading Networks*

Table 6: The effects of the keyphrase candidate filtration steps on the number of positive and negative training instances on the SemEval dataset

Filtration				
Stopword	WordNet	Instances	Positive instances	Negative instances
OFF	OFF	404,967	2,017	402,950
OFF	ON	398,272	2,166	396,106
ON	OFF	223,614	1,949	221,665
ON	ON	217,956	2,095	215,861

Table 7: F-scores achieved on the SemEval dataset by our final model and top-ranked shared task participants

	Reader			Combined			Author		
Method	@5	@10	@15	@5	@10	@15	@5	@10	@15
HUMB	17.8	22.5	23.5	19.8	26.0	27.5	**23.9**	22.2	19.3
Merged$_{BIES+CF}$	**19.0**	21.6	22.2	**20.7**	24.5	25.2	**23.9**	20.0	16.6
WINGNUS	18.0	21.4	22.0	20.5	24.7	25.2	21.0	18.2	14.8
Maui	14.7	16.4	16.1	17.8	20.4	20.6	23.0	19.8	16.2

with Price-Setting Agents – as the official scorer returned a document level F-score of value 0 for the predicted set of Porter-stemmed keyphrases, being *trader, nash equilibrium, game theori, price, network format, buyer, seller, market microstructur, agent, trade, posit profit, price-set agent, bid, mechan design, subgam perfect nash equilibrium*.

For the above document, the expected set of combined (i.e. either reader or author) Porter-stemmed phrases were:
algorithm game theori, market, trade network, interact of buyer and seller, initi endow of monei, bid price, perfect competit, benefit, maximum and minimum amount, econom and financ, strateg behavior of trader, complementari slack, monopoli, trade network.

Inspecting the title or the set of gold standard phrases of the document, the predicted keyphrases – in contrast to the document-level F-score they account for – are arguably not entirely useless.

Missed phrases in the gold standard set were often super-phrases of some predicted phrase or vice versa, *e.g. algorithm game theori* and *game theori* or *market* and *market microstructure*. There were also phrases in the gold standard set, the meaning of which can be composed from distinct elements of the predicted set, such as *bid price* versus *bid* and *price*.

4.2.2 Evaluation on the Inspec dataset

The results achieved by our method on the Inspec dataset alongside with the performance scores of previously published approaches on the same dataset can be found in Table 8 and 9. Table 8 also explicitly states what kind of calculation (i.e. permissive or restrictive) was employed in the previous works

Table 8: Results previously published systems and our model achieves on the Inspec dataset

Method	Evaluation	P	R	F
Hulth [14]	permissive	0.252	0.517	0.339
Merged$_{BIES+CF}$	restrictive	0.281	0.430	0.340
TextRank [28]	restrictive	0.312	0.431	0.362
KeyCluster [21]	permissive	0.350	0.660	0.457

during the calculation of their recall scores. It can be seen that our approach performs competitively to previously published results on that dataset. We should add that approaches on the Inspec dataset tend to achieve high results more easily, as abstracts are typically short, resulting in the phenomenon that a larger proportion of the candidate terms can be useful, compared to the scenario when keyphrases need to be extracted from full documents. This assumption is in concordance with the observations of others, *e.g.* [13, 4]. [4] re-implemented the TextRank algorithm and evaluated it on the SemEval dataset against the combined set of author and reader-assigned keyphrases, which resulted in an F-score of 5.6. Out of the 19 participants of the shared task, 18 achieved better results than that.

Only the KeyCluster approach – based on the clustering of keyphrase candidates as described in [21] – seems to be superior to all other existing frameworks on the Inspec dataset. We should remind the reader, however, that these results were obtained via the permissive calculation of the recall values. Obviously, if the authors reported their evaluation in a restrictive manner, their results would be somewhat lower (yet better than other approaches, but with a much narrower margin) – as also pointed out by [13]. A further concern with respect the KeyCluster algorithm is that its authors report their best performances, when they choose the number of clusters, m, as a function of the keyphrase candidates, n, as either $m = \frac{2}{3}n$ or $m = \frac{4}{5}n$ when performing hierarchical clustering and spectral clustering, respectively. This suggests that it might not be the clustering that is really beneficial in that approach, but the candidate generation step preceding it, as the best results were obtained when the number of clusters were not chosen to be considerably smaller compared to the extracted number of candidate terms. Due to the shortness of the documents in the Inspec dataset, i.e. 136.3 tokens per document on average as reported in [4], this approach could produce effective results. However, the performance of this approach is likely to degrade severely, if it was evaluated on the SemEval dataset consisting of documents having an average length of 5179.6 tokens per document, as also reported in [4].

Indeed, the authors of KeyCluster report in their other work [20] that the clustering-based method performed poorly on long articles (not originating from the SemEval dataset). In their paper, they claim that the Topical PageRank (TPR) algorithm is better at handling longer documents as well. For this reason, we compared their results reported on the Inspec dataset with the performance

Table 9: Comparing our results with that of the Topical PageRank approach on the Inspec dataset

Method	P@5	R@5	F@5	Bpref	MRR
Topical PageRank [20]	0.354	0.183	0.242	0.274	0.583
Merged$_{BIES+CF}$	0.381	0.194	0.257	0.326	0.657

of our system. The effectiveness of the TPR algorithm was characterized by two further measures – besides the precision, recall and F-score –, namely the binary preference measure [5] and the mean reciprocal rank [44]. These measures not only take into consideration the proportion of the correctly determined keyphrases at some given threshold, but also account for the quality of the ranking of keyphrases. The binary preference measure (Bpref) measure is calculated by the formula,

$$bpref = \frac{1}{R}\sum_{r} 1 - \frac{|n \text{ ranked higher than } r|}{R},$$

for a document with R relevant keyphrases, r being a relevant keyphrase of the document and n is a member of the first R non-relevant keyphrases that were returned by a system. Mean-reciprocal rank (MRR) evaluates the quality of the extracted keyphrases by looking at the position of the first correctly extracted keyphrase for each document in the test collection. MRR is then calculated as follows,

$$MRR = \frac{1}{|D|}\sum_{d \in D} \frac{1}{rank_d},$$

where D is the set of test documents and $rank_d$ denotes the rank of the first extracted keyphrase that is also present in the set of gold standard keyphrases with respect document $d \in D$. Our results and those of TPR can be found in Table 9, from which we can see that our approach consistently outperforms the TPR algorithm regarding all the evaluation metrics. In Table 9, P@5, R@5 and F@5 refer to the precision, recall and F-score values measured for the top-5 keyphrases, respectively.

5 Conclusions and further directions

In this paper, a new supervised approach was presented for keyphrase extraction, which introduces novel features making use of extra-textual information. Two means for the integration of external knowledge were presented, namely the usage of Wikipedia for generating multiword expressions-related features and the utilization of the knowledge relying in its category structure and WordNet for the normalization of phrase candidates. The proposed approach is not the first to make use of Wikipedia, but, unlike its predecessors, it does not require a full index on all the textual contents of Wikipedia to be available. Despite not relying on a full index of Wikipedia, our approach was still able to perform

competitively or even better to other similar systems. Furthermore, our proposed method performed consistently well on two standard keyphrase extraction datasets, implying its widespread applicability. An on-line demo and the entire source code of the keyphrase extraction framework proposed in this work can be accessed from the URL `http://rgai1.inf.u-szeged.hu/~berend`.

In the future, we would like to make our system capable of adaptively choose the number of extracted keyphrases per documents and experiment with methods to reduce the extent to which overlapping keyphrases are returned. Furthermore, we wish to experiment with alternative sources of extra-textual information that could be exploited during the extraction of keyphrases.

Acknowledgements

This research was supported by the European Union and the State of Hungary, co-financed by the European Social Fund in the framework of TÁMOP 4.2.4. A/2-11-1-2012-0001 'National Excellence Program'.

References

[1] Ken Barker and Nadia Cornacchia. Using noun phrase heads to extract document keyphrases. In *Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence*, AI '00, pages 40–52, London, UK, UK, 2000. Springer-Verlag.

[2] Gábor Berend. Opinion expression mining by exploiting keyphrase extraction. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1162–1170, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing.

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[4] Adrien Bougouin, Florian Boudin, and Béatrice Daille. TopicRank: Graph-based topic ranking for keyphrase extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 543–551, Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing.

[5] Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 25–32, New York, NY, USA, 2004. ACM.

[6] Alexander Budanitsky and Graeme Hirst. Evaluating WordNet-based measures of lexical semantic relatedness. *Comput. Linguist.*, 32(1):13–47, March 2006.

[7] Zhuoye Ding, Qi Zhang, and Xuanjing Huang. Keyphrase extraction from online news using binary integer programming. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 165–173, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing.

[8] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Comput. Linguist.*, 19(1):61–74, March 1993.

[9] Folke Eisterlehner, Andreas Hotho, and Robert Jäschke, editors. *ECML PKDD Discovery Challenge 2009 (DC09)*, volume 497 of *CEUR-WS.org*, September 2009.

[10] Richárd Farkas, Gábor Berend, István Hegedűs, András Kárpáti, and Balázs Krich. Automatic free-text-tagging of online news archives. In *Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 529–534, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.

[11] C. Fellbaum. *WordNet: An Electronic Lexical Database*. Language, Speech and Communication. Mit Press, 1998.

[12] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *In Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1606–1611, 2007.

[13] Kazi Saidul Hasan and Vincent Ng. 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, pages 365–373, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[14] Anette Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 216–223, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[15] Su Nam Kim and Min-Yen Kan. Re-examining automatic keyphrase extraction approaches in scientific articles. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, MWE '09, pages 9–16, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[16] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. SemEval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 21–26, Morristown, NJ, USA, 2010. ACL.

[17] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. Automatic keyphrase extraction from scientific articles. *Language resources and evaluation*, 47(3):723–742, 2013.

[18] Thomas K Landauer and Susan T. Dutnais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, pages 211–240, 1997.

[19] Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. 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, pages 620–628, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[20] Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 366–376, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[21] Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 257–266, Singapore, August 2009.

[22] Patrice Lopez and Laurent Romary. HUMB: Automatic key term extraction from scientific articles in grobid. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 248–251, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[23] Patrice Lopez, Laurent Romary, et al. GRISP: A massive multilingual terminological database for scientific and technical domains. In *LREC 2010*, 2010.

[24] Abdulhussain E. Mahdi and Arash Joorabchi. A citation-based approach to automatic topical indexing of scientific literature. *J. Inf. Sci.*, 36(6):798–811, December 2010.

[25] Andrew Kachites McCallum. MALLET: A machine learning for language toolkit, 2002. http://mallet.cs.umass.edu.

[26] Olena Medelyan, Eibe Frank, and Ian H. Witten. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1318–1327, Singapore, August 2009. Association for Computational Linguistics.

[27] Olena Medelyan and Ian H. Witten. Thesaurus based automatic keyphrase indexing. In *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '06, pages 296–297, New York, NY, USA, 2006. ACM.

[28] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into texts. In *Proceedings of EMNLP*, volume 4, page 275. Barcelona, Spain, 2004.

[29] Gilad Mishne. AutoTag: a collaborative approach to automated tag assignment for weblog posts. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 953–954, New York, NY, USA, 2006. ACM Press.

[30] Roberto Navigli and Simone Paolo Ponzetto. BabelRelate! a joint multilingual approach to computing semantic relatedness. In *AAAI Conference on Artificial Intelligence*, 2012.

[31] Thuy Dung Nguyen and Min-Yen Kan. Keyphrase extraction in scientific publications. In *Proceedings of the 10th international conference on Asian digital libraries: looking back 10 years and forging new frontiers*, ICADL'07, pages 317–326, Berlin, Heidelberg, 2007. Springer-Verlag.

[32] Thuy Dung Nguyen and Minh-Thang Luong. WINGNUS: Keyphrase extraction utilizing document logical structure. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 166–169, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[33] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web., November 1999. Previous number = SIDL-WP-1999-0120.

[34] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. WordNet::Similarity: Measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL–Demonstrations '04, pages 38–41, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[35] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[36] Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann A. Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for nlp. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '02, pages 1–15, London, UK, UK, 2002. Springer-Verlag.

[37] Sanjay Sood, Sara Owsley, Kristian Hammond, and Larry Birnbaum. TagAssist: Automatic tag suggestion for blog posts. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*, 2007.

[38] Michael Strube and Simone Paolo Ponzetto. WikiRelate! computing semantic relatedness using wikipedia. In *AAAI'06: Proc. of the 21st National Conference on Artificial Intelligence*, pages 1419–1424, 2006.

[39] M. Tatu, M. Srikanth, and T. D'Silva. RSDC'08: Tag recommendations using bookmark content. In *Proceedings of the ECML PKDD Discovery Challenge 2008*, 2008.

[40] Takashi Tomokiyo and Matthew Hurst. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment - Volume 18*, MWE '03, pages 33–40, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[41] Kristina Toutanova and Christopher D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora*, EMNLP '00, pages 63–70, Stroudsburg, PA, USA, 2000. ACL.

[42] Peter Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2:303–336, 2000.

[43] Peter Turney. Coherent keyphrase extraction via web mining. In *Proceedings of IJCAI '03*, pages 434–439, 2003.

[44] Ellen M. Voorhees. The TREC-8 question answering track report. In *In Proceedings of TREC-8*, pages 77–82, 1999.

[45] Xiaojun Wan and Jianguo Xiao. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd national conference on Artificial intelligence - Volume 2*, AAAI'08, pages 855–860. AAAI Press, 2008.

[46] David X. Wang, Xiaoying Gao, and Peter Andreae. DIKEA: Domain-independent keyphrase extraction algorithm. In *Proceedings of the 25th Australasian Joint Conference on Advances in Artificial Intelligence*, AI'12, pages 719–730, Berlin, Heidelberg, 2012. Springer-Verlag.

[47] Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. Kea: Practical automatic keyphrase extraction. In *ACM DL*, pages 254–255, 1999.

[48] Zhaohui Wu and C Lee Giles. Measuring term informativeness in context. In *Proceedings of NAACL-HLT*, pages 259–269, 2013.

[49] Eric Yeh, Daniel Ramage, Christopher D. Manning, Eneko Agirre, and Aitor Soroa. WikiWalk: Random walks on wikipedia for semantic relatedness. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, TextGraphs-4, pages 41–49, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[50] Wei You, Dominique Fontaine, and Jean-Paul A. Barthès. An automatic keyphrase extraction system for scientific documents. *Knowl. Inf. Syst.*, 34(3):691–724, 2013.