# A Domain Independent Double Layered Approach to Keyphrase Generation

Dario De Nart and Carlo Tasso

*Artificial Intelligence Lab, Department of Mathematics and Computer Science, University of Udine, Udine, Italy*

Keywords: Keyphrase Extraction, Keyphrase Inference, Information Extraction, Text Classification, Text Summarization.

Abstract: The annotation of documents and web pages with semantic metatdata is an activity that can greatly increase the accuracy of Information Retrieval and Personalization systems, but the growing amount of text data available is too large for an extensive manual process. On the other hand, automatic keyphrase generation, a complex task involving Natural Language Processing and Knowledge Engineering, can significantly support this activity. Several different strategies have been proposed over the years, but most of them require extensive training data, which are not always available, suffer high ambiguity and differences in writing style, are highly domain-specific, and often rely on a well-structured knowledge that is very hard to acquire and encode. In order to overcome these limitations, we propose in this paper an innovative domain-independent approach that consists of an unsupervised keyphrase extraction phase and a subsequent keyphrase inference phase based on loosely structured, collaborative knowledge such as Wikipedia, Wordnik, and Urban Dictionary. This double layered approach allows us to generate keyphrases that both describe and classify the text.

## 1 INTRODUCTION

The tremendous and constant growth of the amount of text data available on the web has lead, in the last years, to an increasing demand for automatic summarization and information filtering systems. Such systems, in order to be effective and efficient, need metadata capable of representing text contents in a compact, yet detailed way.

As broadly discussed in literature and proven by web usage analysis (Silverstein et al., 1999), is particularly convenient for such metadata to come in the form of *KeyPhrases*(KP), since they can be very expressive (much more than single keywords), straightforward in their meaning, and have a high cognitive plausibility, because humans tend to think in terms of KPs rather than single keywords. In the rest of this paper we will refer to *KP generation* as the process of associating a meaningful set of KPs to a given text, regardless to their origin, while we will call *KP extraction* the act of selecting a set of KP from the text and *KP inference* the act of associating to the text a set of KP that may not be found inside it. KP generation is a trivial and intuitive task for humans, since anyone can tell at least the main topics of a given text, or decide whether it belongs to a certain domain (news item, scientific literature, narrative, etc., ...) or not,

but it can be extremely hard for a machine since most of the documents available lack any kind of semantic hint.

Over the years several authors addressed this issue proposing different approaches towards both KP extraction and inference, but, in our opinion, each one of them has severe practical limitations that prevent massive employment of automatic KP generation in *Information Retrieval*, *Social Tagging*, and *Adaptive Personalization*. Such limitations are the need of training data, the impossibility of associating to a given text keyphrases which are not already included in that text, the high domain specificity, and the need of structured, detailed, and extensive domain knowledge coded in the form of a thesaurus or an ontology. We claim that, in order to match the KP generation performances of a human expert, automatic KP generation systems should both extract and infer KPs, moreover such systems should be unsupervised and domain independent in order to be extensively used, since training data and domain ontologies are hard to obtain.

In order to support our claim we propose here an unsupervised KP generation method that consists of two layers of analysis: a KP Extraction phase and a KP inference one, based on Ontology Reasoning upon knowledge sources that though not being for-

mal ontologies can be seen as loosely structured ones. The first phase provides KPs extracted from the text, describing its content in detail, while the second provides more general KPs, chosen from a controlled dictionary, categorizing the text, rather than describing it.

The rest of the paper is organized as follows: in Section 2 we briefly introduce some related works; in Section 3 we present our keyphrase extraction technique; in Section 4 we illustrate our keyphrase inference technique; in Section 5 we discuss some experimental results and, finally, in Section 6 we conclude the paper.

## 2 RELATED WORK

Many works over the past few years have discussed different solutions for the problem of automatically tagging documents and Web pages as well as the possible applications of such technologies in the fields of Personalization and Information Retrieval in order to significantly reduce information overload and increase accuracy. Both keyphrase extraction and inference have been widely discussed in literature. Several different keyphrase extraction techniques have been proposed, which usually are structured into two phases:

- a *candidate phrase identification* phase, in which all the possible phrases are detected in the text;

- a *selection* phase in which only the most significant of the above phrases are chosen as keyphrases.

The wide span of proposed methods can be roughly divided into two distinct categories:

- *Supervised approaches*: the underlying idea of these methods is that KP Extraction can be seen as a *classification* problem and therefore solved with a sufficient amount of training data (manually annotated) and machine learning algorithms (Turney, 2000). Several authors addressed the problem in this direction (Turney, 1999) and many systems that implement supervised approaches are available, such as KEA (Witten et al., 1999), Extractor[2], and LAKE (DAvanzo et al., 2004). All the above systems can be extremely effective and, as far as reliable data sets are available, can be flawlessly applied to any given domain (Marujo et al., 2013), however requiring training data in order to work properly, implies two major drawbacks: (i) the quality of the extraction process relies on the quality of training data and (ii) a model trained on a specific domain just won't fit another application domain unless is trained again.

- *Unsupervised approaches*: this second class of methods eliminates the need for training data by selecting candidate KP according to some ranking strategy. Most of the proposed systems rely on the identification of *noun phrases*, i.e. phrases made of just nouns and then proceed with a further selection based on heuristics such as frequency of the phrase (Barker and Cornacchia, 2000) or upon phrase clustering (Bracewell et al., 2005). A third approach proposed by (Mihalcea and Tarau, 2004) and (Litvak and Last, 2008), exploits a graph-based ranking model algorithm, bearing much similarity to the notorious Page Rank algorithm, in order to select significant KPs and identify related terms that can be summarized by a single phrase. All the above techniques share the same advantage over the supervised strategies, that is being truly domain independent, since they rely on general principles and heuristics and therefore there is no need for training data.

Hybrid approaches have been proposed as well, incorporating semi-supervised domain knowledge in an otherwise unsupervised extraction strategy (Sarkar, 2013), but still remain highly domain-specific.

Keyphrase extraction, however, is severely limited by the fact it can ultimately return only words contained in the input document, which are highly prone to ambiguity and subject to the nuances of different writing styles (e.g: an author can write "mining frequent patterns" where another one would write "frequent pattern mining" ). Keyphrase inference can overcome these limitations and has been widely explored in literature as well, spanning from systems that simply combine words appearing in the text in order to construct rather than extract phrases (Danilevsky et al., 2013) to systems that assign KPs that may built with terms that never appear in the document. In the latter case, KPs come from a controlled dictionary, possibly an ontology; in such case, a classifier is trained in order to find which entries of the exploited dictionary may fit the text (Dumais et al., 1998). If the dictionary of possible KPs is an ontology, its structure can be exploited in order to provide additional evidence for inference (Pouliquen et al., 2006) and, by means of ontological reasoning, evaluate relatedness between terms (Medelyan and Witten, 2006). In (Pudota et al., 2010) is discussed a KP inference technique based on a very specific domain ontology, written in the OWL language, in the context of a vast framework for personalized document annotation that combines both KP Extraction and inference. KP inference based on dictionaries, however, is strongly limited by the size, the domain coverage, and the specificity level of the considered dictionary.

# 3 SYSTEM OVERVIEW

In order to test our approach and to support our claims we developed a new version of the system presented in (Pudota et al., 2010). We introduce a new double-layered architecture and an original innovation, i.e. the exploitation of a number of generalist online External Knowledge Sources, rather than a formal domain specific ontology, in order to improve extraction quality, to infer meaningful KPs not included in the input text and to preserve domain independence.

In Figure 1 the overall organization of the proposed system is presented. It is constituted by the following main components:

- A *KP Extraction Module* (*KPEM*), devoted to analyse the text end extract from it meaningful KPs. It is supported by some linguistic resources, such as a *POS tagger* (for the English Language) and a *Stopwords Database* and it accesses some online *External Knowledge Sources* (*EKSs*) mainly exploited in order to provide support to the candidate KPs identified in the text (as explained in the following section). The KPEM receives in input an unstructured text and it produces in output a ranked list of KPs, which is stored in an *Extracted Keyphrases Data Base* (*EKPDB*).

- A *KP Inference Module* (*KPIM*), which works on the KP list produced by the KPEM and it is devoted to infer new KPs, not already included in the input text. It relies on some ontological reasoning based on the access to the External Knowledge Sources, exploited in order to identify new concepts which are related to the ones referred to by the KPs previously extracted by the KPEM. Inferred KPs are stored in the *Inferred KP Data Base* (*IKPDB*).

The access to the online External Knowledge Sources is provided by a *Generalized Knowledge Gateway (GKG)*. The system is organized in the form of a Web service, allowing easy access to the KP Generation service to all kinds of clients.

The workflow of the system is intended as a simulation of the typical cognitive process that happens when we are asked to summarize or classify a text. At the beginning all of the text is read, then the KPEM identifies and ranks concepts included in the text, finally, the KPIM precesses the identified concepts in order to infer from them other concepts that may be tightly related or implied. The result of the process is a set of KPs that appear or do not appear in the text, thus mixing explicit and tacit knowledge.

# 4 PHRASE EXTRACTION

KPEM is an enhanced version of *DIKPE*, the unsupervised, domain independent KP extraction approach described in (Pudota et al., 2010) and (Ferrara and Tasso, 2013). In a nutshell, DIKPE generates a large set of candidate KPs; the exploited approach then merges different types of knowledge in order to identify meaningful concepts in a text, also trying to model a human-like KP assignment process. In particular we use: *Linguistic Knowledge* (POS tagging, sentence structure, punctuation); *Statistical Knowledge* (frequency, tf/idf,...); knowledge about the *structure* of a document (position of the candidate KP in the text, title, subtitles, ...); *Meta-knowledge* provided by the author (html tags,...); knowledge coming from *online external knowledge sources*, useful for validating candidate keyphrases which have been socially recognized, for example, in collaborative wikis (e.g. Wikipedia, Wordnik, and other online resources).

By means of the above knowledge sources, each candidate phrase, is characterized by a set of features, such as, for example:

- *Frequency:* the frequency of the phrase in the text;

- *Phrase Depth:* at which point of the text the phrase occurs for the first time: the sooner it appears, the higher the value;

- *Phrase Last Occurrence:* at which point of the text the phrase occurs for the last time: the later it appears, the higher the value;

- *Life Span:* the fraction of text between the first and the last occurrence of the phrase;

- *POS Value:* a parameter taking into account the grammatical composition of the phrase, excluding some patterns and assigning higher priority to other patterns (typically, for example but not exclusively, it can be relevant to consider the number of nouns in the phrase over the number of words in the phrase).

- *WikiFlag*: a parameter taking into account the fact that the phrase is or is not an entry of online collaborative external knowledge sources (EKSs); the WikiFlag provides evidence of the social meaningfulness for a KP and therefore can be considered a feature based on general knowledge.

A weighted linear combination of the above features, called *Keyphraseness* is then computed and the KPs are sorted in descending keyphraseness order. The weight of each feature can be tuned in order to fit particular kinds of text, but, usually, a generalist preset can be used with good results. The topmost n KPs are finally suggested.
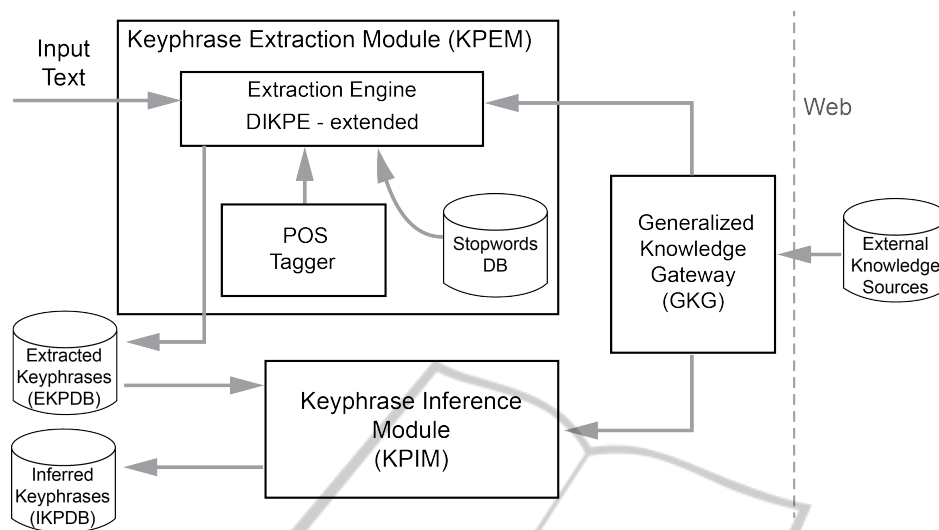
Figure 1: Architecture of the System.

In this work, we extended the DIKPE system with the GKG module, allowing access to multiple knowledge sources at the same time. We also added a more general version of the WikiFlag feature. This feature is computed as follows: if the phrase matches an entry in at least one of the considered external knowledge sources, then its value is set to 1, otherwise the phrase is split into its constituents and the WikiFlag value is set to the percentage corresponding to the number of terms that have a match in at least one of the considered external knowledge sources. By doing so, a KP that does not match as phrase, but is constituted by terms that match as single words, still gets a high score, but lower than a KP that features a perfect match. The WikiFlag feature is processed as all the other features, contributing to the computation of the keyphraseness and therefore influencing the ranking of the extracted KPs. The rationale of this choice is that a KP is important insofar it represents a meaningful concept or entity, rather than a random combination of words, and matching a whole phrase against a collaborative human-made knowledge source (as the EKSs are) guarantees that it makes better sense, providing a strong form of human/social validation. However, the WikiFlag does not prevent terms and phrases that are not validated by external knowledge to be suggested as KPs if they appear with significant frequency in significant parts of the document, which may be the case of newly introduced terminology or highly specific jargon. Exploiting the Wikiflag actually helps in reducing the tendency of the system to suggest typos, document parsing errors, random combinations of frequent non-stopwords terms, and other kinds of false positives.

Another improvement over the original DIKPE approach is represented by the fact that, instead of suggesting the top n KPs extracted, the new system evaluates the decreasing trend of Keyphraseness among ordered KPs, it detects the first significant downfall (detected by evaluation of the derivative function) in the keyphraseness value, and it suggests all the KPs occurring before that (dynamic) threshold. By doing so, the system suggests a variable number of high-scored KPs, while the previous version suggests a fixed number of KPs, that could have been either too small or too large for the given text.

## 5 PHRASE INFERENCE

The KP Inference Module (KPIM), as well as the knowledge-based WikiFlag feature described in the previous section, rely on a set of external knowledge sources that are accessed via web. In the following we call *entity* any entry present in one or more EKSs; entities may have a complex structure, as well as include different kinds of data (e.g: text, pictures, videos, ...), however we are interested in the relationships occurring between entities rather than their content. EKs may be online databases, such as Wordnet, linked data or traditional web resources as long as a dense link structure with some well-recognizable semantics is available. We assume that (i) there is a way to match extracted KPs with entities described in EKSs (e.g.: querying the exploited service using the KP as search key) and (ii) each one of the EKSs considered is organized according to some kind of hierarchy. Such hierarchy may be loose, but it must include some kind of *is-a* and *is-related* relationships, allowing us to infer,

308

for each entity, a set of parent and a set of related entities. Such sets may be void, since we do not assume each entity being necessarily linked to at least another one, nor the existence of a root entity that is ancestor of all the other entities in the ontology.

Even if such structure is loose, assuming its existence is nowadays not trivial at all; however, along with the growth of semantic web resources, an increasing number of collaborative resources allow users to classify and link together knowledge items, generating an increasing number of pseudo-ontologies. Clear examples of this trend are Wikipedia, where almost any article contains links to other articles and many articles are grouped into *categories*, and Wordnik, an online collaborative dictionary where any word is associated to a set of hypernyms, synonyms, hyponyms, and related terms. Recently also several entertainment sites, like Urban Dictionary, have begun to provide these possibilities, making them eligible knowledge sources for our approach. Knowledge sources may be either generalist (like Wikipedia), or specific (like the many domain-specific wikis hosted on *wikia.com*) and several different EKSs can be exploited at the same time in order to provide better results.

In the case of Wikipedia, parent entities are given by the *categories*, that are thematic groups of articles (i.e.: "Software Engineering" belongs to the "Engineering Disciplines" category). An entry may belong to several categories, for example the entry on "The Who" belongs to the "musical quartets" category as well as to the "English hard rock musical groups" one and the "Musical groups established in 1964" one. Related entities, instead, can be derived from links contained in the page associated to the given entity: such links can be very numerous and heterogeneous, but the most closely related ones are often grouped into one or more *templates*, that are the thematic collections of internal Wikipedia links usually displayed on the bottom of the page, as shown in Figure 2. For instance, in a page concerning a film director, it is very likely to find a template containing links to the all movies he directed or the actors he worked with.

Wordnik, instead, provides hierarchical information explicitly by associating to any entity lists of hypernyms (parent entities) and synonyms (related entities).

The inference algorithm considers the topmost half of the extracted KPs, that typically is still a significantly larger set than the one presented as output, and, for each KP that can be associated to an entity, retrieves from each EKS a set of parent entities and a set of related entities. If a KP corresponds to more than one entity on one or more EKSs, all of the retrieved

entities are taken into account. The sets associated to single KPs are then merged into a table of related entities and a table of parent entities for the whole text. Each retrieved entity is scored accordingly to the sum of the Keyphraseness value of the KPs from which it has been derived and then it is sorted by descending score. The top entries of such tables are suggested as meaningful KPs for the input document.

By doing so, we select only entities which are related to or parent of a significant number of hi-scored KPs, addressing the problem of polysemy among the extracted KP. For instance, suppose we extracted "Eiffel" and "Java Language" from the same text: they both are polysemic phrases since the first may refer to a ISO-standardized OO language as well as to a French civil engineer and architect and the latter to the programming language or to the language spoken in the island of Java. However, since they appear together, and they are both part of the "Object-Oriented Programming Languages" category in Wikipedia, it can be deduced that the text is about computer science rather than architecture or Indonesian languages.

## 6 EVALUATION

Formative tests were performed in order to test the accuracy of the inferred KPs and their ability to add meaningful information to the set of extracted KPs, regardless of the domain covered by the input text. Several data sets, dealing with different topics, were processed, article by article, with the same feature weights and exploiting Wikipedia and Wordnik as External Knowledge Source. For each article a list of extracted KPs and one of inferred KPs were generated, then the occurrences of each KP were counted, in order to evaluate which portion of the data set is covered by each KP. We call *set coverage* the fraction of the data set labelled with a single KP. Since the topics covered in the texts included in each data set are known a-priori, we expect the system to generate KPs that associate the majority of the texts in the data set to their specific domain topic.

The first data set contained 113 programming tutorials, spanning from brief introductions published on blogs and forums to extensive articles taken from books and journals, covering both practical and theoretical aspects of programming. A total of 776 KPs were extracted and 297 were inferred. In Table 1 are reported the most frequently extracted and inferred KPs. As expected, extracted KPs are highly specific and tend to characterize a few documents in the set (the most frequent KP covers just the 13% of the data set), while inferred ones provide an higher level
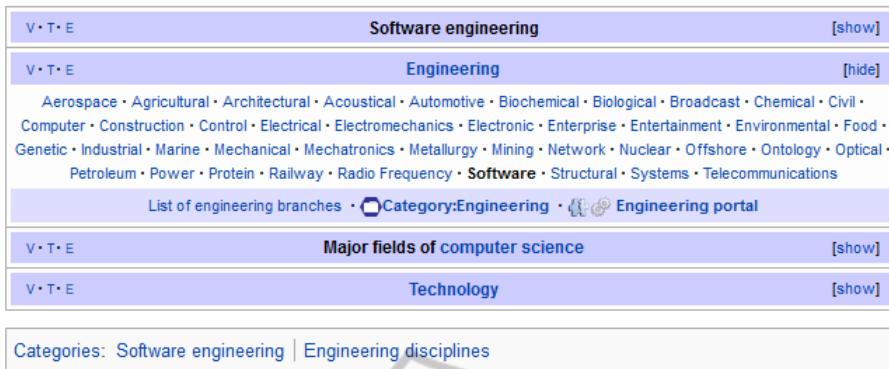
Figure 2: The lowest section of a Wikipedia page, containing templates (the "Engineering" template has been expanded) and categories (bottom line).

Table 1: The most frequently extracted and inferred KPs from the "programming tutorials" data set.

| Extracted Keyphrase | Set coverage | Inferred Keyphrase | Set Coverage |
|---|---|---|---|
| program | 0,13 | Mathematics | 0,47 |
| use | 0,12 | Programming language | 0,26 |
| function | 0,12 | move | 0,25 |
| type | 0,10 | Computer science | 0,22 |
| programming language | 0,10 | Set (mathematics) | 0,17 |
| programming | 0,08 | Data types | 0,15 |
| functions | 0,07 | Aristotle | 0,16 |
| class | 0,07 | Function (mathematics) | 0,14 |
| code | 0,06 | C (programming language) | 0,14 |
| COBOL | 0,06 | Botanical nomenclature | 0,12 |
| chapter | 0,05 | C++ | 0,11 |
| variables | 0,05 | Information | 0,08 |
| number | 0,05 | Java (programming language) | 0,08 |

Table 2: The most frequently extracted and inferred KPs from the "album reviews" data set.

| Extracted Keyphrase | Set coverage | Inferred Keyphrase | Set Coverage |
|---|---|---|---|
| metal | 0,23 | Music genre | 1 |
| album | 0,21 | Record label | 0,97 |
| death metal | 0,17 | Record producer | 0,54 |
| black metal | 0,17 | United States | 0,48 |
| band | 0,16 | Studio album | 0,16 |
| bands | 0,08 | United Kingdom | 0,11 |
| death | 0,08 | Bass guitar | 0,09 |
| old school | 0,07 | Single (music) | 0,08 |
| sound | 0,06 | Internet Movie Database | 0,07 |
| albums | 0,05 | Heavy metal music | 0,07 |
| power metal | 0,05 | Allmusic | 0,06 |

of abstraction, resulting in an higher coverage over the considered data set. However some Inferred KPs are not accurate, such as " Botanical nomenclature " that clearly derive from the presence of terms such as "tree", "branch", "leaf", and "forest" that are frequently used in Computer Science, and "Aristotele" which comes from the frequent references to Logic,

which Wikipedia frequently associates with the Greek philosopher.

Another data set contained reviews of 211 heavy metal albums published in 2013. Reviews were written by various authors, both professionals and non-professionals, and combine a wide spectrum of writing styles, from utterly specific, almost scientific, to

highly sarcastic, with many puns and popular culture references.

In Table 2 are reported the most frequently extracted and inferred KPs. All the documents in the set were associated with the Inferred KP "Music Genre" and the 97% of them with "Record Label", which clearly associates the texts with the music domain. Evaluation and development are still ongoing and new knowledge sources, such as domain-specific wikis and Urban Dictionary, are being considered.

## 7 CONCLUSIONS AND FUTURE WORK

In this paper we proposed a truly domain independent approach to both KP extraction and inference, able to generate significant semantic metadata with two different layers of abstraction (phrase extraction and phrase inference) for any given text without need for training. The KP extraction part of the system provides a very fine level of detail, producing KPs that may not be found in a controlled dictionary (such as Wikipedia), but characterize the text. Such KPs are extremely valuable for the purpose of summarization and provide great accuracy when used as search keys. However, they are not widely shared, meaning, from an information retrieval point of view, a very poor recall. On the other hand, the KP inference part generates only KPs taken from a controlled dictionary (the union of the considered EKSs) that are more likely to be general, widely known and used, and, therefore, shared among a significant number of texts.

As shown in the previous section, our approach can annotate a set of documents with meaningful KPs, however, a few unrelated KPs may be inferred, mostly due to ambiguities of the text and to the generalist nature of the exploited online external knowledge sources. This unrelated terms, fortunately, tend to appear in a limited number of cases and to be clearly unrelated not only to the majority of the generated KPs, but also to each other. In fact, our next step in this research will be precisely to identify such false positives by means of an estimate of the *Semantic Relatedness* (Strube and Ponzetto, 2006), (Ferrara and Tasso, 2012) between terms in order to identify, for each generated KP, a list of related concepts and detect concept clusters in the document.

The proposed KP generation technique can be applied both in the Information Retrieval domain and in the Adaptive Personalization one. The previous version of the DIKPE system has already been integrated with good results in RES (De Nart et al., 2013), a personalized content-based recommender system for sci-entific papers that suggests papers accordingly to their similarity with one or more documents marked as interesting by the user, and in the PIRATES framework (Pudota et al., 2010) for tag recommendation and automatic document annotation. We expect this extended version of the system to provide an even more accurate and complete KP generation and, therefore, to improve the performance of these existing systems, in this way supporting the creation of new Semantic Web Intelligence tools.

## REFERENCES

Barker, K. and Cornacchia, N. (2000). Using noun phrase heads to extract document keyphrases. In *Advances in Artificial Intelligence*, pages 40–52. Springer.

Bracewell, D. B., Ren, F., and Kuriowa, S. (2005). Multilingual single document keyword extraction for information retrieval. In *Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on*, pages 517–522. IEEE.

Danilevsky, M., Wang, C., Desai, N., Guo, J., and Han, J. (2013). Kert: Automatic extraction and ranking of topical keyphrases from content-representative document titles. *arXiv preprint arXiv:1306.0271*.

DAvanzo, E., Magnini, B., and Vallin, A. (2004). Keyphrase extraction for summarization purposes: The lake system at duc-2004. In *Proceedings of the 2004 document understanding conference*.

De Nart, D., Ferrara, F., and Tasso, C. (2013). Personalized access to scientific publications: from recommendation to explanation. In *User Modeling, Adaptation, and Personalization*, pages 296–301. Springer.

Dumais, S., Platt, J., Heckerman, D., and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 148–155. ACM.

Ferrara, F. and Tasso, C. (2012). Integrating semantic relatedness in a collaborative filtering system. In *Mensch & Computer Workshopband*, pages 75–82.

Ferrara, F. and Tasso, C. (2013). Extracting keyphrases from web pages. In *Digital Libraries and Archives*, pages 93–104. Springer.

Litvak, M. and Last, M. (2008). Graph-based keyword extraction for single-document summarization. In *Proceedings of the workshop on multi-source multilingual information extraction and summarization*, pages 17–24. Association for Computational Linguistics.

Marujo, L., Gershman, A., Carbonell, J., Frederking, R., and Neto, J. P. (2013). Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization. *arXiv preprint arXiv:1306.4886*.

Medelyan, O. and Witten, I. H. (2006). Thesaurus based automatic keyphrase indexing. In *Proceedings of the 6th*

*ACM/IEEE-CS joint conference on Digital libraries*, pages 296–297. ACM.

Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into texts. In *Proceedings of EMNLP*, volume 4. Barcelona, Spain.

Pouliquen, B., Steinberger, R., and Ignat, C. (2006). Automatic annotation of multilingual text collections with a conceptual thesaurus. *arXiv preprint cs/0609059*.

Pudota, N., Dattolo, A., Baruzzo, A., Ferrara, F., and Tasso, C. (2010). Automatic keyphrase extraction and ontology mining for content-based tag recommendation. *International Journal of Intelligent Systems*, 25(12):1158–1186.

Sarkar, K. (2013). A hybrid approach to extract keyphrases from medical documents. *arXiv preprint arXiv:1303.1441*.

Silverstein, C., Marais, H., Henzinger, M., and Moricz, M. (1999). Analysis of a very large web search engine query log. In *ACm SIGIR Forum*, volume 33, pages 6–12. ACM.

Strube, M. and Ponzetto, S. P. (2006). Wikirelate! computing semantic relatedness using wikipedia. In *AAAI*, volume 6, pages 1419–1424.

Turney, P. D. (1999). Learning to extract keyphrases from text. national research council. *Institute for Information Technology, Technical Report ERB-1057*.

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

Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., and Nevill-Manning, C. G. (1999). Kea: Practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries*, pages 254–255. ACM.