KP-Miner: Participation in SemEval-2

Samhaa R. El-Beltagy
Cairo University
Giza, Egypt.
samhaa@computer.org

Ahmed Rafea
The American University in Cairo
New Cairo, Egypt.
rafea@aucegypt.edu

Abstract

This paper briefly describes the KP-Miner system which is a system developed for the extraction of keyphrases from English and Arabic documents, irrespective of their nature. The paper also outlines the performance of the system in the "Automatic Keyphrase Extraction from Scientific Articles" task which is part of SemEval-2.

1 Introduction

KP-Miner (El-Beltagy, 2006) (El-Beltagy, 2009) is a system for the extraction of keyphrases from English and Arabic documents. When developing the system, the goal was to build a general purpose keyphrase extraction system that can be easily configured by users based on their understanding of the documents from which keyphrases are to be extracted and without the need for any training documents or the use of any sophisticated natural language processing or linguistic tools. As such, the keyphrase extraction process in KP-Miner is an un-supervised one. When building a general purpose keyphrase extraction system, this was an important objective, as training data is not always readily available for any type of data. The goal of entering the KP-Miner system into the SemEval-2 competition, was to see how well it will perform on a specific task, without making any changes in its default parameters.

2 System Overview

Keyphrase extraction in the KP-Miner system is a three step process: candidate keyphrase selection, candidate keyphrase weight calculation and finally keyphrase refinement. Each of these steps, is explained in the following sub-sections. More details about the employed algorithm, and justification for using certain values for selected parameters, can be found in (El-Beltagy, 2009).

2.1 Candidate keyphrase selection

In KP-Miner, a set of rules is employed in order to elicit candidate keyphrases. As a phrase will never be separated by punctuation marks within some given text and will rarely have stop words within it, the first condition a sequence of words has to display in order to be considered a candidate keyphrase, is that it is not be separated by punctuation marks or stop words. A total of 187 common stopwords (the, then, in, above, etc) are used in the candidate keyphrase extraction step. After applying this first condition on any given document, too many candidates will be generated; some of which will make no sense to a human reader. To filter these out, two further conditions are applied. The first condition states that a phrase has to have appeared at least ***n*** times in the document from which keyphrases are to be extracted, in order to be considered a candidate keyphrase. This is called the least allowable seen frequency(lasf) factor and in the English version of the system, this is set to 3. However, if a document is short, ***n*** is decremented depending on the length of the document.

The second condition is related to the position where a candidate keyphrase first appears within an input document. Through observation as well as experimentation, it was found that in long documents, phrases occurring for the first time after a given threshold, are very rarely keyphrases. So a cutoff constant CutOff is defined in terms of a number of words after which if a phrase appears for the first time, it is filtered out and ignored. The initial prototype of the KP-Miner system (El-Beltagy, 2006), set this cutoff value to a constant (850). Further experimentation carried out in (El-Beltagy, 2009) revealed that an optimum value for this constant is 400. In

Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 190–193,
Uppsala, Sweden, 15-16 July 2010.

the implementation of the KP-Miner system, the phrase extraction step described above is carried out in two phases. In the first phase, words are scanned until either a punctuation mark or a stop word is encountered. The scanned sequence of words and all possible n-grams within the encountered sequence where ***n*** can vary from 1 to sequence length-1, are stemmed and stored in both their original and stemmed forms. If the phrase (in its stemmed or original form) or any of its sub-phrases, has been seen before, then the count of the previously seen term is incremented by one, otherwise the previously unseen term is assigned a count of one. Very weak stemming is performed in this step using only the first step of the Porter stemmer (Porter, 1980). In the second phase, the document is scanned again for the longest possible sequence that fulfills the conditions mentioned above. This is then considered as a candidate keyphrase. Unlike most of the other keyphrase extraction systems, the devised algorithm places no limit on the length of keyphrases, but it was found that extracted keyphrases rarely exceed three terms.

2.2 Candidate keyphrases weight calculation

Single key features obtained from documents by models such as TF-IDF (Salton and Buckley, 1988) have already been shown to be representative of documents from which they've been extracted as demonstrated by their wide and successful use in clustering and classification tasks. However, when applied to the task of keyphrase extraction, these same models performed very poorly (Turney, 1999). By looking at almost any document, it can be observed that the occurrences of phrases is much less frequent than the occurrence of single terms within the same document. So it can be concluded that one of the reasons that TF-IDF performs poorly on its own when applied to the task of keyphrase extraction, is that it does not take this fact into consideration which results in a bias towards single words as they occur in larger numbers. So, a boosting factor is needed for compound terms in order to balance this bias towards single terms. In this work for each input document ***d*** from which keyphrases are to be extracted, a boosting factor B_d is calculated as follows:

$B_d = |N_d| / (|P_d| * \propto)$

and if $B_d > \sigma$ then $B_d = \sigma$

Here $|N_d|$ is the number of all candidate terms in document **d**, $|P_d|$ is the number of candidate terms whose length exceeds one in document ***d*** and $\propto$ and σ are weight adjustment constants. The values used by the implemented system are 3 for σ and 2.3 for $\propto$.

To calculate the weights of document terms, the TF-IDF model in conjunction with the introduced boosting factor, is used. However, another thing to consider when applying TF-IDF for a general application rather than a corpus specific one, is that keyphrase combinations do not occur as frequently within a document set as do single terms. In other words, while it is possible to collect frequency information for use by a general single keyword extractor from a moderately large set of random documents, the same is not true for keyphrase information. There are two possible approaches to address this observation. In the first, a very large corpus of a varied nature can be used to collect keyphrase related frequency information. In the second, which is adopted in this work, any encountered phrase is considered to have appeared only once in the corpus. This means that for compound phrases, frequency within a document as well as the boosting factor are really what determine its weight as the idf value for all compound phrases will be a constant ***c*** determined by the size of the corpus used to build frequency information for single terms. If the position rules described in (El-Beltagy, 2009) are also employed, then the position factor is also used in the calculation for the term weights. In summary, the following equation is used to calculate the weight of candidate keyphrases whether single or compound:

$$w_{ij} = tf_{ij} * idf * B_i * P_f$$

Where:

w_{ij} = weight of term t_j in Document D_i

tf_{ij} = frequency of term t_j in Document D_i

idf = log_2 N/n where N is the number of documents in the collection and n is number of documents where term t_j occurs at least once. If the term is compound, n is set to 1.

B_i = *the boosting factor associated with document* D_i

P_f = *the term position associated factor. If position rules are not used this is set to 1.*

2.3 Final Candidate Phrase List Refinement

The KP-Miner system, allows the user to specify a number ***n*** of keyphrases s/he wants back and uses the sorted list to return the top ***n*** keyphrases requested by the user. The default number of ***n*** is five. As stated in step one, when

generating candidate keyphrases, the longest possible sequence of words that are uninterrupted by possible phrase terminators, are sought and stored and so are sub-phrases contained within that sequence provided that they appear somewhere in the text on their own. For example, if the phrase 'excess body weight' is encountered five times in a document, the phrase itself will be stored along with a count of five. If the sub-phrase , 'body weight', is also encountered on its own, than it will also be stored along with the number of times it appeared in the text including the number of times it appeared as part of the phrase 'excess body weight'. This means that an overlap between the count of two or more phrases can exist. Aiming to eliminate this overlap in counting early on can contribute to the dominance of possibly noisy phrases or to overlooking potential keyphrases that are encountered as sub-phrases. However, once the weight calculation step has been performed and a clear picture of which phrases are most likely to be key ones is obtained, this overlap can be addressed through refinement. To refine results in the KP-Miner system, the top ***n*** keys are scanned to see if any of them is a sub-phrase of another. If any of them are, then its count is decremented by the frequency of the term of which it is a part. After this step is completed, weights are recalculated and a final list of phrases sorted by weight, is produced. The reason the top ***n*** keys rather than all candidates, are used in this step is so that lower weighted keywords do not affect the outcome of the final keyphrase list. It is important to note that the refinement step is an optional one, but experiments have shown that in the English version of the system, omitting this step leads to the production of keyphrase lists that match better with author assigned keyword. In (El-Beltagy, 2009) the authors suggested that employing this step leads to the extraction of higher quality keyphrases. Experimentation carried out on the Gold standard dataset provided by the organizers of the SemEval-2 competition on "Keyphrase Extraction from Scientific Documents" and described in the next section, seems to suggest that this idea is a valid one.

3 Participation in the SemEval-2 Competition

One of the new tracks introduced to SemEval this year is a track dedicated entirely to keyphrase extraction from scientific articles. The task was proposed with the aim of providing participants with "the chance to compete and benchmark" this technology (SemEval2, 2010).

In this competition, participants were provided with 40 trial documents, 144 training documents, and 100 test documents. For the trial and training data, three sets of answers were provided: author-assigned keyphrases, reader-assigned keyphrases, and finally a set that is simply a combination between the 2 previous sets. Unlike author-assigned keyphrases, which may or may not occur in the content, all reader-assigned keyphrases were said to have been extracted from the papers. The participants were then asked to produce the top 15 keyphrases for each article in the test document set and to submit the stemmed version of these to the organizers.

Evaluation was carried out in the traditional way in which keyphrase sets extracted by each of the participants were matched against answer sets (i.e. author-assigned keyphrases and reader-assigned keyphrases) to calculate precision, recall and F-score. Participants were then ranked by F-score when extracting all 15 keyphrases.

Since the KP-miner system is an unsupervised keyphrase extraction system, no use was made of the trial and training data. The system was simply run of the set of test documents, and the output was sent to the organizers. 2 different runs were submitted: one produced used the initial prototype of the system, (El-Beltagy, 2006), while the second was produced using the more mature version of the system (El-Beltagy, 2009). Both systems were run without making any changes to their default parameters. The idea was to see how well the KP-Miner would fair among other keyphrase extraction systems without any additional configuration. The more mature version of the system performed better when its results were compared to the author-reader combined keyphrase set and consequently was the one whose final results were taken into consideration in the competition. The system ranked at 2 , with a tie between it and another system when extracting 15 keyphrases from the combined keypharse set. The results are shown in table 1.

	Precision	**Recall**	**F-Score**
HUMB	27.2%	27.8%	27.5%
WINGNUS	24.9%	25.5%	25.2%
KP-Miner	**24.9%**	**25.5%**	**25.2%**
SZTERGAK	24.8%	25.4%	25.1%
ICL	24.6%	25.2%	24.9%
SEERLAB	24.1%	24.6%	24.3%

KX_FBK	23.6%	24.2%	23.9%
DERIUNLP	22.0%	22.5%	22.3%
Maui	20.3%	20.8%	20.6%
DFKI	20.3%	20.7%	20.5%
BUAP	19.0%	19.4%	19.2%
SJTULTLAB	18.4%	18.8%	18.6%
UNICE	18.3%	18.8%	18.5%
UNPMC	18.1%	18.6%	18.3%
JU_CSE	17.8%	18.2%	18.0%
LIKEY	16.3%	16.7%	16.5%
UvT	14.6%	14.9%	14.8%
NIRAJIIITH	14.1%	14.5%	14.3%
POLYU	13.9%	14.2%	14.0%
UKP	5.3%	5.4%	5.3%

Table 1: Performance of all participating systems over combined keywords when extracting 15 keyphrases

When evaluating the system on reader assigned keyphrases only (again when extracting 15 keyphrases), the KP-Miner system ranked at 6 with a tie between it and another system. The system's precision, recall, and f-score were: 19.3%, 24.1% , 21.5% respectively.

To test whether the phrase refinement step described in section 2.3 would improve the results or not, this option was turned on, and the results were evaluated using the script and the golden dataset provided by the competition organizers. The results are shown in tables 2 and 3.

	Precision	**Recall**	**F-Score**
Top 5	29.6%	12.3%	17.4%
Top 10	23.3%	20.5%	24.3%
Top 15	25.3%	26.1%	25.8%

Table 2: Performance over combined keywords when extracting, 5, 10, and 15 keyphrases

	Precision	**Recall**	**F-Score**
Top 5	37.8%	12.9%	19.2%
Top 10	30.3%	19.4%	21.1%
Top 15	20.1%	25.1%	22.3%

Table 3: Performance over reader assigned keywords when extracting, 5, 10, and 15 keyphrases

Had these results been submitted, the system would have still ranked at number 2 (but more comfortably so) when comparing its results to the combined author-reader set of keywords, but it would jumped to third place for the reader assigned keyphrases. This improvement confirms what the authors hypothesized in (El-Beltagy, 2009) which is that the usage of the final refinement step does lead to better quality keyphrases.

4 Conclusion and future work

Despite the fact that the KP-Miner was designed as a general purpose keyphrase extraction system, and despite the simplicity of the system and the fact that it requires no training to function, it seems to have performed relatively well when carrying out the task of keyphrase extraction from scientific documents. The fact that it was outperformed, seems to indicate that for optimal performance for this specific task, further tweaking of the system's parameters should be carried out. In future work, the authors will investigate the usage of machine learning techniques for configuring the system for specific tasks. A further improvement to the system can entail allowing certain stopwords to appear within the produced keyphrases. It is worth noting that the organizers stated that 55 of the reader assigned keyphrases and 6 of the author assigned keyphrases (making a total of 61 keyphrases in the combined dataset), contained the "of" stopword. However, none of these would have been detected by the KP-Miner system as currently "of" is considered as a phrase terminator.

References

M Porter. 1980. An Algorithm for Suffix Stripping, *Program*, 14, 130-137.

G. Salton and C. Buckley. 1988. Term-weighting Approaches in Automatic Text Retrieval, *Information Processing and Management,* 24:513-523.

Peter D. Turney. 1999. Learning to Extract Keyphrases from Text, *National Research Counc*il, Institute for Information Technology, ERB-1057.

Samhaa. R. El-Beltagy and Ahmed Rafea. 2009. KP-Miner: A Keyphrase Extraction System for English and Arabic Documents *Information Systems*, 34(1):132-144.

Samhaa R. El-Beltagy. 2006. KP-Miner: A Simple System for Effective Keyphrase Extraction. *Proceeding of the 3rd IEEE International Conference on Innovations in Information Technology (IIT '06)*, Dubai, UAE.

SemEval. 2010. http://semeval2.fbk.eu/semeval2.php