BUAP: An Unsupervised Approach to Automatic Keyphrase Extraction from Scientific Articles

Roberto Ortiz, David Pinto, Mireya Tovar
Faculty of Computer Science, BUAP
Puebla, Mexico
korn_resorte2003@hotmail.com,
{dpinto, mtovar}@cs.buap.mx

Héctor Jiménez-Salazar
Information Technologies Dept., UAM
DF, Mexico
hgimenezs@gmail.com

Abstract

In this paper, it is presented an unsupervised approach to automatically discover the latent keyphrases contained in scientific articles. The proposed technique is constructed on the basis of the combination of two techniques: maximal frequent sequences and pageranking. We evaluated the obtained results by using micro-averaged precision, recall and F-scores with respect to two different gold standards: 1) reader's keyphrases, and 2) a combined set of author's and reader's keyphrases. The obtained results were also compared against three different baselines: one unsupervised (TF-IDF based) and two supervised (Naïve Bayes and Maximum Entropy).

1 Introduction

The task of automatic keyphrase extraction has been studied for several years. Firstly, as semantic metadata useful for tasks such as summarization (Barzilay and Elhadad, 1997; Lawrie et al., 2001; DAvanzo and Magnini, 2005), but later recognizing the impact that good keyphrases would have on the quality of various Natural Language Processing (NLP) applications (Frank et al., 1999; Witten et al., 1999; Turney, 1999; Barker and Corrnacchia, 2000; Medelyan and Witten, 2008). Thus, the selection of important, topical phrases from within the body of a document may be used in order to improve the performance of systems dealing with different NLP problems such as, clustering, question-answering, named entity recognition, information retrieval, etc.

In general, a keyphrase may be considered as a sequence of one or more words that capture the main topic of the document, as that keyphrase is expected to represent one of the key ideas expressed by the document author. Following the previously mentioned hypothesis, we may take advantage of two different techniques of text analysis: maximal frequent sequences to extract a sequence of one or more words from a given text, and pageranking, expecting to extract those word sequences that represent the key ideas of the author.

The interest on extracting high quality keyphrases from raw text has motivated forums, such as SemEval, where different systems may evaluate their performances. The purpose of SemEval is to evaluate semantic analysis systems. In particular, in this paper we are reporting the results obtained in Task #5 of SemEval-2 2010, which has been named: "Automatic Keyphrase Extraction from Scientific Articles". We focused this paper on the description of our approach and, therefore, we do not describe into detail the task nor the dataset used. For more information about this information read the "Task #5 Description paper", also published in this proceedings volume (Nam Kim et al., 2010).

The rest of this paper is structured as follows. Section 2 describes into detail the components of the proposed approach. In Section 3 it is shown the performance of the presented system. Finally, in Section 4 a discussion of findings and further work is given.

2 Description of the approach

The approach presented in this paper relies on the combination of two different techniques for selecting the most prominent terms of a given text: maximal frequent sequences and pageranking. In Figure 1 we may see this two step approach, where we are considering a sequence to be equivalent to an n-gram. The complete description of the procedure is given as follows.

We select maximal frequent sequences which

Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 174–177,
Uppsala, Sweden, 15-16 July 2010. ©2010 Association for Computational Linguistics

we consider to be candidate keyphrases and, thereafter, we ranking them in order to determine which ones are the most importants (according to the pageranking algorithm). In the following subsections we give a brief description of these two techniques. Afterwards, we provide an algorithm of the presented approach.

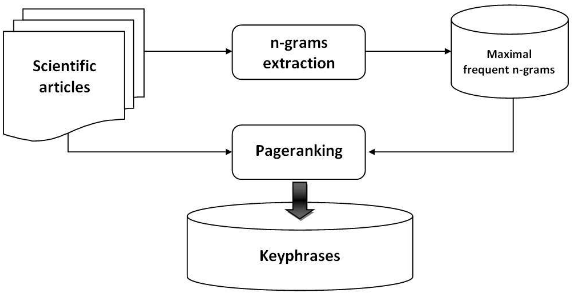

Figure 1: Two step approach of BUAP Team at the Task #5 of SemEval-2

2.1 Maximal Frequent Sequences

Definition: If a sequence p is a subsequence of q and the number of elements in p is equal to n, then the p is called an n-gram in q.

Definition: A sequence $p = a_1 \cdots a_k$ is a subsequence of a sequence q if all the items a_i occur in q and they occur in the same order as in p. If a sequence p is a subsequence of a sequence q we say that p occurs in q.

Definition: A sequence p is frequent in S if p is a subsequence of at least β documents in S where β is a given frequency threshold. Only one occurrence of sequence in the document is counted. Several occurrences within one document do not make the sequence more frequent.

Definition: A sequence p is a maximal frequent sequence in S if there does not exists any sequence q in S such that p is a subsequence of q and p is frequent in S.

2.2 PageRanking

The algorithm of PageRanking was defined by Brin and Page in (Brin and Page, 1998). It is a graph-based algorithm used for ranking webpages. The algorithm considers input and output links of each page in order to construct a graph, where each vertex is a webpage and each edge may be the input or output links for this webpage. They denote as $In(V_i)$ the set of input links of webpage V_i, and $Out(V_i)$ their output links. The algorithm proposed to rank each webpage based on the voting or recommendation of other webpages. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex. Moreover, the importance of the vertex casting the vote determines how important the vote itself is, and this information is also taken into account by the ranking model.

Although this algoritm has been initially proposed for webpages ranking, it has been also used for other NLP applications which may model their corresponding problem in a graph structure. Eq. (1) is the formula proposed by Brin and Page.

$$S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j) \quad (1)$$

where d is a damping factor that can be set between 0 and 1, which has the role of integrating into the model the probability of jumping from a given vertex to another random vertex in the graph. This factor is usually set to 0.85 (Brin and Page, 1998).

There are some other propossals, like the one presented in (Mihalcea and Tarau, 2004), where a textranking algorithm is presented. The authors consider a weighted version of PageRank and present some applications to NLP using unigrams. They also construct multi-word terms by exploring the conections among ranked words in the graph. Our algorithm differs from textranking in that we use MFS for feeding the PageRanking algorithm.

2.3 Algorithm

The complete algoritmic description of the presented approach is given in Algorithm 1. Readers and writers keyphrases may be quite different. In particular, writers usually introduce acronyms in their text, but they use the complete or expanded representation of these acronyms for their keyphrases. Therefore, we have included a module ($Extract\_Acronyms$) for extracting both, acronyms with their corresponding expanded version, which are used afterwards as output of our system. We have preprocessed the dataset removing stopwords and punctuation symbols. Lemmatization (TreeTagger[1]) and stemming (Porter Stemmer (Porter, 1980)) were also applied in some stages of preprocessing.

The $Maximal\_Freq\_Sequences$ module extracts maximal frequent sequences of words and we feed the PageRaking module ($PageRanking$)

[1] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

with all these sequences for determining the most important ones. We use the structure of the scientific articles in order to determine in and out links of the sequences found. In fact, we use a neighborhood criterion (a pair of MFS in the same sentence) for determining the links between those MFS's. Once the ranking is calculated, we may select those sequences of a given length (unigrams, bigrams and trigrams) as output of our system. We also return a maximum of three acronyms, and their associated multiterm phrases ($MultiTerm$), as candidate keyphrases. Determining the length and quantity of the sequences (n-grams) was experimentally deduced from the training corpus.

Algorithm 1: Algorithm of the Two Step approach for the Task #5 at SemEval-2

```
Input: A document set: $D = \{d_1, d_2, \cdots\}$
Output: A set $K = \{K_1, K_2, \cdots\}$ of
        keyphrases for each document $d_i$:
        $K_i = \{k_{i,1}, k_{i,2}, \cdots\}$
foreach $d_i \in D$ do
    $AcronymSet$ = Extract_Acronyms($d_i$);
    $d_i^1$ = Pre_Processing($d_i$);
    $MFS$ = Maximal_Freq_Sequences($d_i^1$);
    $CK$ = PageRanking($d_i^1$, $MFS$);
    $CU$ = Top_Nine_Unigrams($CK$);
    $CT$ = Top_Three_Trigrams($CK$);
    $K_i = CT$;
    $NU = 0$;
    $Acronyms = 0$;
    foreach $unigram \in CU$ do
        if $unigram \in AcronymSet$ then
            if $Acronyms < 3$ then
                $K_i = K_i \bigcup \{unigram\}$;
                $EA$ = MultiTerm($unigram$);
                $K_i = K_i \bigcup \{EA\}$;
                $Acronyms$++;
            end
        else
            $K_i = K_i \bigcup \{unigram\}$;
            $NU$++;
        end
    end
    $N = (15-(2*Acronyms+|CT|+NU))$;
    $CB$ = Top_N_Bigrams($CK$, $N$);
    $K_i = K_i \bigcup CB$;
end
return $K = \{K_1, K_2, \cdots\}$
```

In this edition of the Task #5 of SemEval-2 2010, we tested three different runs, which were named: $BUAP-1$, $BUAP-2$ and $BUAP-3$. Definition and differences among the three runs are given in Table 3.

The results obtained with each run, together with three different baselines are given in the following section.

3 Experimental results

In all tables, P, R, F mean micro-averaged precision, recall and F-scores. For baselines, there were provided 1,2,-3 grams as candidates and $TFIDF$ as features. In Table 2, $TFIDF$ is an unsupervised method to rank the candidates based on $TFIDF$ scores. NB and ME are supervised methods using Naïve Bayes and maximum entropy in WEKA. In second column, R means to use the reader-assigned keyword set as gold-standard data and C means to use both author-assigned and reader-assigned keyword sets as answers.

Notice from Tables 2 and 3 that we outperformed all the baselines for the Top 15 candidates. However, the Top 10 candidates were only outperformed by the *Reader*-Assigned keyphrases found. This implies that the *Writer* keyphrases we obtained were not of as good as the *Reader* ones. As we mentioned, readers and writers assign different keywords. The former write keyphrases based on the lecture done, by the latter has a wider context and their keyphrases used to be more complex. We plan to investigate this issue in the future.

4 Conclusions

We have presented an approach based on the extraction of maximal frequent sequences which are then ranked by using the pageranking algorithm. Three different runs were tested, modifying the preprocessing stage and the number of bigrams given as output. We did not see an improvement when we used lemmatization of the documents. The run which obtained the best results was ranking by the organizer according to the top 15 best keyphrases, however, we may see that our runs need to be analysed more into detail in order to provide a re-ranking procedure for the best 15 keyphrases found. This procedure may improve the top 5 candidates precision.

Run name		*Description*
$BUAP-1$	:	This run is exactly the one described in Algorithm 1.
$BUAP-2$	:	Same as $BUAP-1$ but lemmatization was applied a priori and stemming at the end.
$BUAP-3$	:	Same as $BUAP-2$ but output twice the number of bigrams.

Table 1: Description of the three runs submitted to the Task #5 of SemEval-2 2010

Method	by	top 5 candidates			top 10 candidates			top 15 candidates		
		P	R	F	P	R	F	P	R	F
$TF-IDF$	R	17.80%	7.39%	10.44%	13.90%	11.54%	12.61%	11.60%	14.45%	12.87%
	C	22.00%	7.50%	11.19%	17.70%	12.07%	14.35%	14.93%	15.28%	15.10%
NB	R	16.80%	6.98%	9.86%	13.30%	11.05%	12.07%	11.40%	14.20%	12.65%
	C	21.40%	7.30%	10.89%	17.30%	11.80%	14.03%	14.53%	14.87%	14.70%
ME	R	16.80%	6.98%	9.86%	13.30%	11.05%	12.07%	11.40%	14.20%	12.65%
	C	21.40%	7.30%	10.89%	17.30%	11.80%	14.03%	14.53%	14.87%	14.70%

Table 2: Baselines

Method	by	top 5 candidates			top 10 candidates			top 15 candidates		
		P	R	F	P	R	F	P	R	F
$BUAP-1$	R	10.40%	4.32%	6.10%	13.90%	11.54%	12.61%	14.93%	18.60%	16.56%
	C	13.60%	4.64%	6.92%	17.60%	12.01%	14.28%	19.00%	19.44%	19.22%
$BUAP-2$	R	10.40%	4.32%	6.10%	13.80%	11.46%	12.52%	14.67%	18.27%	16.27%
	C	14.40%	4.91%	7.32%	17.80%	12.14%	14.44%	18.73%	19.17%	18.95%
$BUAP-3$	R	10.40%	4.32%	6.10%	12.10%	10.05%	10.98%	12.33%	15.37%	13.68%
	C	14.40%	4.91%	7.32%	15.60%	10.64%	12.65%	15.67%	16.03%	15.85%

Table 3: The three different runs submitted to the competition

Acknowledgments

This work has been partially supported by CONACYT (Project #106625) and PROMEP (Grant #103.5/09/4213).

References

[Barker and Corrnacchia2000] K. Barker and N. Cornacchia. 2000. Using noun phrase heads to extract document keyphrases. In *13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence*.

[Barzilay and Elhadad1997] R. Barzilay and M. Elhadad. 1997. Using lexical chains for text summarization. In *ACL/EACL 1997 Workshop on Intelligent Scalable Text Summarization*, pages 10–17.

[Brin and Page1998] S. Brin and L. Page. 1998. The anatomy of a large-scale hypertextual web search engine. In *COMPUTER NETWORKS AND ISDN SYSTEMS*, pages 107–117. Elsevier Science Publishers B. V.

[DAvanzo and Magnini2005] E. DAvanzo and B. Magnini. 2005. A keyphrase-based approach to summarization:the lake system. In *Document Understanding Conferences (DUC-2005)*.

[Frank et al.1999] E. Frank, G.W. Paynter, I. Witten, C. Gutwin, and C.G. Nevill-Manning. 1999. Domain specific keyphrase extraction. In *16th International Joint Conference on AI*, pages 668–673.

[Lawrie et al.2001] D. Lawrie, W. B. Croft, and A. Rosenberg. 2001. Finding topic words for hierarchical summarization. In *SIGIR 2001*.

[Medelyan and Witten2008] O. Medelyan and I. H. Witten. 2008. Domain independent automatic keyphrase indexing with small training sets. *Journal of American Society for Information Science and Technology*, 59(7):1026–1040.

[Mihalcea and Tarau2004] R. Mihalcea and P. Tarau. 2004. Textrank: Bringing order into texts. In *EMNLP 2004, ACL*, pages 404–411.

[Nam Kim et al.2010] S. Nam Kim, O. Medelyan, and M.Y. Kan. 2010. Semeval-2010 task5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the Fifth International Workshop on Semantic Evaluations (SemEval-2010)*. Association for Computational Linguistics.

[Porter1980] M. F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3).

[Turney1999] P. Turney. 1999. Learning to extract keyphrases from text. Technical Report ERB-1057. (NRC #41622), National Research Council, Institute for Information Technology.

[Witten et al.1999] I. Witten, G. Paynter, E. Frank, C. Gutwin, and G. Nevill-Manning. 1999. Kea:practical automatic key phrase extraction. In *fourth ACM conference on Digital libraries*, pages 254–256.