SZTERGAK : Feature Engineering for Keyphrase Extraction

Gábor Berend
Department of Informatics
University of Szeged
2. Árpád tér Szeged, H-6720, Hungary
berendg@inf.u-szeged.hu

Richárd Farkas
Hungarian Academy of Sciences
103. Tisza Lajos körút
Szeged, H-6720, Hungary
rfarkas@inf.u-szeged.hu

Abstract

Automatically assigning keyphrases to documents has a great variety of applications. Here we focus on the keyphrase extraction of scientific publications and present a novel set of features for the supervised learning of keyphraseness. Although these features are intended for extracting keyphrases from scientific papers, because of their generality and robustness, they should have uses in other domains as well. With the help of these features SZTERGAK achieved top results on the *SemEval-2 shared task on Automatic Keyphrase Extraction from Scientific Articles* and exceeded its baseline by 10%.

1 Introduction

Keyphrases summarize the content of documents with the most important phrases. They can be valuable in many application areas, ranging from information retrieval to topic detection. However, since manually assigned keyphrases are rarely provided and creating them by hand would be costly and time-consuming, their automatic generation is of great interest nowadays. Recent state-of-the-art systems treat this kind of task as a supervised learning task, in which phrases of a document should be classified with respect to their key phrase characteristics based on manually labeled corpora and various feature values.

This paper focuses on the task of keyphrase extraction from scientific papers and we shall introduce new features that can significantly improve the overall performance. Although the experimental results presented here are solely based on scientific articles, due to the robustness and universality of the features, our approach is expected to achieve good results when applied on other domains as well.

2 Related work

In **keyphrase extraction** tasks, phrases are extracted from one document that are the most characteristic of its content (Liu et al., 2009; Witten et al., 1999). In these approaches keyphrase extraction is treated as a classification task, in which certain n-grams of a specific document act as keyphrase candidates, and the task is to classify them as proper keyphrases or not.

While Frank et al. (1999) exploited domain specific knowledge to improve the quality of automatic tagging, others like Liu et al. (2009) analyze term co-occurence graphs. It was Nguyen and Kan (2007) who dealt with the special characteristics of scientific papers and introduced the state-of-the-art feature set to keyphrase extraction tasks. Here we will follow a similar approach and make significant improvements by the introduction of novel features.

3 The SZTERGAK system

The SZTERGAK framework treats the reproduction of reader-assigned keyphrases as a supervised learning task. In our setting a restricted set of token sequences extracted from the documents was used as classification instances. These instances were ranked regarding to their posteriori probabilities of the `keyphrase` class, estimated by a Naïve Bayes classifier. Finally, we chose the top-15 candidates as keyphrases.

Our features can be grouped into four main categories: those that were calculated solely from the surface characteristics of phrases, those that took into account the document that contained a keyphrase, those that were obtained from the given document set and those that were based on external sources of information.

Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 186–189,
Uppsala, Sweden, 15-16 July 2010.

3.1 Preprocessing

Since there are parts of a document (e.g. tables or author affiliations) that can not really contribute to the keyphrase extractor, several preprocessing steps were carried out. Preprocessing included the elimination of author affiliations and messy lines.

The determination of the full title of an article would be useful, however, it is not straightforward because of multi-line titles. To solve this problem, a web query was sent with the first line of a document and its most likely title was chosen by simply selecting the most frequently occurring one among the top 10 responses provided by the Google API. This title was added to the document, and all the lines before the first occurrence of the line `Abstract` were omitted.

Lines unlikely to contain valuable information were also excluded from the documents. These lines were identified according to statistical data of their surface forms (e.g. the average and the deviation of line lengths) and regular expressions. Lastly, section and sentence boundaries were found in a rule-based way, and the POS and syntactic tagging (using the Stanford parser (Klein and Manning, 2003)) of each sentence were carried out.

When syntactically parsed sentences were obtained, keyphrase aspirants were extracted. The 1 to 4-long token sequences that did not start or end with a stopword and consisted only of POS-codes of an `adjective`, a `noun` or a `verb` were defined to be possible keyphrases (resulting in classification instances). Tokens of key phrase aspirants were stemmed to store them in a uniform way, but they were also appended by the POS-code of the derived form, so that the same root forms were distinguished if they came from tokens having different POS-codes, like there shown in Table 1.

Textual Appearance	Canonical form
regulations	regul_nns
Regulation	regul_nn
regulates	regul_vbz
regulated	regul_vbn

Table 1: Standardization of document terms.

3.2 The extended feature set

The features characterizing the extracted keyphrase aspirants can be grouped into four main types, namely phrase-, document-, corpus-level and external knowledge-based features. Below we will describe the different types of features as well as those of KEA (Witten et al., 1999) which are cited as default features by most of the literature dealing with keyphrase extraction.

3.2.1 Standard features

Features belonging to this set contain those of KEA, namely Tf-idf and the first occurrence.

The **Tf-idf feature** assigns the tf-idf metric to each keyphrase aspirant.

The **first occurrence feature** contains the relative first position for each keyphrase aspirant. The feature value was obtained by dividing the absolute first token position of a phrase by the number of tokens of the document in question.

3.2.2 Phrase-level features

Features belonging to this group were calculated solely based on the keyphrase aspirants themselves. Such features are able to get the general characteristics of phrases functioning as keyphrases.

Phrase length feature contains the number of tokens a keyphrase aspirant consists of.

POS feature is a nominal one that stores the POS-code sequence of each keyphrase aspirant. (For example, for the phrase `full_JJ space_NN` its value was `JJ NN`.)

Suffix feature is a binary feature that stores information about whether the original form of a keyphrase aspirant finished with some specific ending according to a subset of the Michigan Sufficiency Exams' Suffix List. [1]

3.2.3 Document-level features

Since keyphrases should summarize the particular document they represent, and phrase-level features introduced above were independent of their context, document-level features were also invented.

Acronymity feature functions as a binary feature that is assigned a true value iff a phrase is likely to be an extended form of an acronym in the same document. A phrase is treated as an extended form of an acronym if it starts with the same letter as the acronym present in its document and it also contains all the letters of the acronym in the very same order as they occur in the acronym.

PMI feature provides a measure of the multiword expression nature of multi-token phrases,

[1] http://www.michigan-proficiency-exams.com/suffix-list.html

and it is defined in Eq. (1), where $p(t_i)$ is the document-level probability of the occurrence of ith token in the phrase. This feature value is a generalized form of pointwise mutual information for phrases with an arbitrary number of tokens.

$$pmi(t_1, t_2, ..., t_n) = \frac{log(\frac{p(t_1,t_2,...,t_n)}{p(t_1)\cdot p(t_2)\cdot ....\cdot p(t_n)})}{log(p(t_1, t_2, ..., t_n))^{n-1}} \quad (1)$$

Syntactic feature values refer to the average minimal normalized depth of the NP-rooted parse subtrees that contain a given keyphrase aspirant at the leaf nodes in a given document.

3.2.4 Corpus-level features

Corpus-level features are used to determine the relative importance of keyphrase aspirants based on a comparison of corpus-level and document-level frequencies.

The **sf-isf feature** was created to deal with logical positions of keyphrases and the formula shown in Eq. (2) resembles that of tf-idf scores (hence its name, i.e. Section Frequency-Inverted Section Frequency). This feature value favors keyphrase aspirants k that are included in several sections of document d (sf), but are present in a relatively small number of sections in the overall corpus (isf). Phrases with higher sf-isf scores for a given document are those that are more relevant with respect to that document.

$$sfisf(k, d) = sf(k, d) * isf(k) \quad (2)$$

Keyphraseness feature is a binary one which has a true value iff a phrase is one of the 785 different author-assigned keyphrases provided in the training and test corpora.

3.2.5 External knowledge-based features

Apart from relying on the given corpus, further enhancements in performance can be obtained by relying on external knowledge sources.

Wikipedia-feature is assigned a true value for keyphrase aspirants for which there exists a Wikipedia article with the same title. Preliminary experiments showed that this feature is noisy, thus we also investigated a relaxed version of it, where occurrences of Wikipedia article titles were looked for only in the title and abstract of a paper.

Besides using Wikipedia for feature calculation, it was also utilized to retrieve semantic orientations of phrases. Making use of **redirect links of Wikipedia**, the semantic relation of synonymity can be exploited. For example, as there exists a redirection between Wikipedia articles `XML` and `Extensible Markup Language`, it may be assumed that these phrases mean the same. For this reason during the training phase we treated a phrase equivalent to its redirected version, i.e. if there is a keyphrase aspirant that is not assigned in the gold-standard reader annotation but the Wikipedia article with the same title has a redirection to such a phrase that is present among positive keyphrase instances of a particular document, the original phrase can be treated as a positive instance as well. In this way the ratio of positive examples could be increased from 0.99% to 1.14%.

Feature combinations	**F-score**
Standard features (SF)	14.57
SF + phrase length feature	20.93
SF + POS feature	19.60
SF + suffix feature	16.35
SF + acronymity feature	16.87
SF + PMI feature	15.68
SF + syntactic feature	14.20
SF + sf-isf feature	14.79
SF + keyphraseness feature	15.17
SF + Wikipedia feature - full paper	14.37
SF + Wikipedia feature - abstract	16.50
SF + Wikipedia redirect	14.50
Shared Task best baseline	12.87
All features	23.82
All features - keyphraseness excluded	22.11

Table 2: Results obtained with different features.

4 Results and discussion

The training and test sets of the shared task (Kim et al., 2010) consisted of 144 and 100 scientific publications from the ACL repository, respectively. Since the primary evaluation of the shared task was based on the top-15 ranked automatic keyphrases compared to the keyphrases assigned by the readers of the articles, these results are reported here. The evaluation results can be seen in Table 2 where the individual effect of each feature is given in combination with the standard features.

It is interesting to note the improvement obtained by extending standard features with the simple feature of phrase length. This indicates that though the basic features were quite good, they did not take into account the point that reader

keyphrases are likely to consist of several words.

Morphological features, such as POS or suffix features were also among the top-performing ones, which seems to show that most of the keyphrases tend to have some common structure. In contrast, the syntactic feature made some decrease in the performance when it was combined just with the standard ones. This can be due to the fact that the input data were quite noisy, i.e. some inconsistencies arose in the data during the pdf to text conversion of articles, which made it difficult to parse some sentences correctly.

It was also interesting to see that Wikipedia feature did not improve the result when it was applied to the whole document. However, our previous experiences on keyphrase extraction from scientific abstracts showed that this feature can be very useful. Hence, we relaxed the feature to handle occurrences just from the abstract. This modification of the feature yielded a 14.8% improvement in the F-measure. A possible explanation for this is that Wikipedia has articles of very common phrases (such as `Calculation` or `Result`) and the distribution of such non-keyphrase terms is higher in the body of the articles than in abstracts.

The last row of Table 2 contains the result achieved by the complete feature set excluding *keyphraseness*. As keyphraseness exploits author-assigned keyphrases and – to the best of our knowledge – other participants of the shared task did not utilize author-assigned keyphrases, this result is present in the final ranking of the shared task systems. However, we believe that if the task is to extract keyphrases from an article to gain semantic meta-data for an NLP application (e.g. for information retrieval or summarization), author-assigned keyphrases are often present and can be very useful. This latter statement was proved by one of our experiments where we used the author keyphrases assigned to the document itself as a binary feature (instead of using the pool of all keyphrases). This feature set could achieve an F-score of 27.44 on the evaluation set and we believe that this should be the complete feature set in a real-world semantic indexing application.

5 Conclusions

In this paper we introduced a wide set of new features that are able to enhance the overall performance of supervised keyphrase extraction applications. Our features include those calculated simply on surface forms of keyphrase aspirants, those that make use of the document- and corpus-level environment of phrases and those that rely on external knowledge. Although features were designed to the specific task of extracting keyphrases from scientific papers, due to their generality it is highly assumable that they can be successfully utilized on different domains as well.

The features we selected in SZTERGAK performed well enough to actually achieve the third place on the shared task by excluding the keyphraseness feature and would be the first by using any author-assigned keyphrase-based feature. It is also worth emphasizing that we think that there are many possibilities to further extend the feature set (e.g. with features that take the semantic relatedness among keyphrase aspirants into account) and significant improvement could be achievable.

Acknowledgement

The authors would like to thank the annotators of the shared task for the datasets used in the shared task. This work was supported in part by the NKTH grant (project codename TEXTREND).

References

Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. In *Proceeding of 16th IJCAI*, pages 668–673.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. Semeval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *Proc. of the 5th SIGLEX Workshop on Semantic Evaluation*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430.

Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on EMNLP*.

Thuy Dung Nguyen and Minyen Kan. 2007. Keyphrase extraction in scientific publications. In *Proc. of International Conference on Asian Digital Libraries (ICADL 07)*, pages 317–326.

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