HUMB: Automatic Key Term Extraction from Scientific Articles in GROBID

Patrice Lopez, Laurent Romary

HUMB: Automatic Key Term Extraction from Scientific Articles in GROBID

Patrice Lopez
INRIA
Berlin, Germany
patrice_lopez@hotmail.com

Laurent Romary
INRIA & HUB-IDSL
Berlin, Germany
laurent.romary@inria.fr

Abstract

The Semeval task 5 was an opportunity for experimenting with the key term extraction module of GROBID, a system for extracting and generating bibliographical information from technical and scientific documents. The tool first uses GROBID's facilities for analyzing the structure of scientific articles, resulting in a first set of structural features. A second set of features captures content properties based on phraseness, informativeness and keywordness measures. Two knowledge bases, GRISP and Wikipedia, are then exploited for producing a last set of lexical/semantic features. Bagged decision trees appeared to be the most efficient machine learning algorithm for generating a list of ranked key term candidates. Finally a post ranking was realized based on statistics of co-usage of keywords in HAL, a large Open Access publication repository.

1 Introduction

Key terms (or keyphrases or keywords) are metadata providing general information about the content of a document. Their selection by authors or readers is, to a large extent, subjective which makes automatic extraction difficult. This is, however, a valuable exercise, because such key terms constitute good topic descriptions of documents which can be used in particular for information retrieval, automatic document clustering and classification. Used as subject headings, better keywords can lead to higher retrieval rates of an article in a digital library.

We view automatic key term extraction as a subtask of the general problem of extraction of technical terms which is crucial in technical and scientific documents (Ahmad and Collingham, 1996). Among the extracted terms for a given scientific document in a given collection, which key terms best characterize this document?

This article describes the system realized for the Semeval 2010 task 5, based on GROBID's (**G**ene**R**ation **O**f **BI**bilographic **D**ata) module dedicated to key term extraction. GROBID is a tool for analyzing technical and scientific documents, focusing on automatic bibliographical data extraction (header, citations, etc.) (Lopez, 2009) and structure recognition (section titles, figures, etc).

As the space for the system description is very limited, this presentation focuses on key aspects. We present first an overview of our approach, then our selection of features (section 3), the different tested machine learning models (section 4) and the final post-ranking (section 5). We briefly describe our unsuccessful experiments (section 6) and we conclude by discussing future works.

2 Bases

Principle As most of the successful works for keyphrase extraction, our approach relies on Machine Learning (ML). The following steps are applied to each document to be processed:

1. Analysis of the structure of the article.
2. Selection of candidate terms.
3. Calculation of features.
4. Application of a ML model for evaluating each candidate term independently.
5. Final re-ranking for capturing relationships between the term candidates.

For creating the ML model, steps 1-3 are applied to the articles of the training set. We view steps 1 and 5 as our main novel contributions. The structure analysis permits the usage of reliable features in relation to the logical composition of the article to be processed. The final re-ranking exploits

general relationships between the set of candidates which cannot be captured by the ML models.

Candidate term selection In the following, *word* should be understood as similar to *token* in the sense of MAF[1]. Step 2 has been implemented in a standard manner, as follows:

1. Extract all n-grams up to 5 words,
2. Remove all candidate n-grams starting or ending with a stop word,
3. Filter from these candidates terms having mathematical symbols,
4. Normalize each candidate by lowercasing and by stemming using the Porter stemmer.

Training data The task's collection consists of articles from the ACM (Association for Computational Machinery) in four narrow domains (*C.2.4* Distributed Systems, *H.3.3* Information Search and Retrieval, *I.2.6* Learning and *J.4* Social and Behavioral Sciences). As training data, we used this task's training resources (144 articles from ACM) and the National University of Singapore (NUS) corpus[2] (156 ACM articles from all computing domains). Adding the additional NUS training data improved our final results (+7.4% for the F-score at top 15, i.e. from 25.6 to 27.5).

3 Features

3.1 Structural features

One of the goals of GROBID is to realize reliable conversions of technical and scientific documents in PDF to fully compliant TEI[3] documents. This conversion implies first the recognition of the different sections of the document, then the extraction of all header metadata and references. The analysis is realized in GROBID with Conditional Random Fields (CRF) (Peng and McCallum, 2004) exploiting a large amount of training data. We added to this training set a few ACM documents manually annotated and obtained a very high performance for field recognitions, between 97% (section titles, reference titles) and 99% (title, abstract) accuracy for the task's collection.

Authors commonly introduce the main concepts of a written communication in the header (title, abstract, table of contents), the introduction, the section titles, the conclusion and the reference list. Similarly human readers/annotators typically focus their attention on the same document parts. We introduced thus the following 6 binary features characterizing the position of a term with respect to the document structure for each candidate: present in the *title*, in the *abstract*, in the *introduction*, in at least one *section titles*, in the *conclusion*, in at least one *reference or book title*.

In addition, we used the following standard feature: the *position of the first occurrence*, calculated as the number of words which precede the first occurrence of the term divided by the number of words in the document, similarly as, for instance, (Witten et al., 1999).

3.2 Content features

The second set of features used in this work tries to captures distributional properties of a term relatively to the overall textual content of the document where the term appears or the collection.

Phraseness The phraseness measures the lexical cohesion of a sequence of words in a given document, i.e. the degree to which it can be considered as a phrase. This measure is classically used for term extraction and can rely on different techniques, usually evaluating the ability of a sequence of words to appear as a stable phrase more often than just by chance. We applied here the Generalized Dice Coeficient (GDC) as introduced by (Park et al., 2002), applicable to any arbitrary n-gram of words ($n \geq 2$). For a given term T, $\mid T \mid$ being the number of words in T, $freq(T)$ the frequency of occurrence of T and $freq(w_i)$ the frequency of occurrence of the word w_i, we have:

$$GDC(T) = \frac{\mid T \mid log_{10}(freq(T))freq(T)}{\sum_{w_i \in T} freq(w_i)}$$

We used a default value for a single word, because, in this case, the association measure is not meaningful as it depends only on the frequency.

Informativeness The *informativeness* of a term is the degree to which the term is representative of a document given a collection of documents. Once again many measures can be relevant, and we opt for the standard TF-IDF value which is used in most of the keyphrase extraction systems, see for instance (Witten et al., 1999) or (Medelyan and

[1]Morpho-syntactic Annotation Framework, see http://pauillac.inria.fr/ clerger/MAF/

[2]http://wing.comp.nus.edu.sg/downloads/keyphraseCorpus

[3]Text Encoding Initiative (TEI), http://www.tei-c.org.

Witten, 2008). The TF-IDF score for a Term T in document D is given by:

$$\text{TF-IDF}(T, D) = \frac{freq(T, D)}{\mid D \mid} \times -log_2 \frac{count(T)}{N}$$

where $\mid D \mid$ is the number of words in D, $count(T)$ is the number of occurrence of the term T in the global corpus, and N is the number of documents in the corpus.

Keywordness Introduced by (Witten et al., 1999), the keywordness reflects the degree to which a term is selected as a keyword. In practice, it is simply the frequency of the keyword in the global corpus. The efficiency of this feature depends, however, on the amount of training data available and the variety of technical domains considered. As the training set of documents for this task is relatively large and narrow in term of technical domains, this feature was relevant.

3.3 Lexical/Semantic features

GRISP is a large scale terminological database for technical and scientific domains resulting from the fusion of terminological resources (MeSH, the Gene Ontology, etc.), linguistic resources (part of WordNet) and part of Wikipedia. It has been created for improving patent retrieval and classification (Lopez and Romary, 2010). The assumption is that a phrase which has been identified as controlled term in these resources tend to be a more important keyphrase. A binary feature is used to indicate if the term is part of GRISP or not.

We use Wikipedia similarly as the *Wikipedia keyphraseness* in Maui (Medelyan, 2009). The *Wikipedia keyphraseness* of a term T is the probability of an appearance of T in a document being an anchor (Medelyan, 2009). We use Wikipedia Miner[4] for obtaining this value.

Finally we introduced an additional feature commonly used in keyword extraction, the *length* of the term candidate, i.e. its number of words.

4 Machine learning model

We experimented different ML models: Decision tree (C4.5), Multi-Layer perceptron (MLP) and Support Vector Machine (SVM). In addition, we combined these models with boosting and bagging techniques. We used WEKA (Witten and Frank, 2005) for all our experiments, except for SVM where LIBSVM (Chang and Lin, 2001) was used. We failed to obtain reasonable results with SVM. Our hypothesis is that SVM is sensitive to the very large number of negative examples compared to the positive ones and additional techniques should be used for balancing the training data. Results for decision tree and MLP were similar but the latter is approx. 57 times more time-consuming for training. Bagged decision tree appeared to perform constantly better than boosting (+8,4% for F-score). The selected model for the final run was, therefore, bagged decision tree, similarly as, for instance, in (Medelyan, 2009).

5 Post-ranking

Post-ranking uses the selected candidates as a whole for improving the results, while in the previous step, each candidate was selected independently from the other. If we have a ranked list of term T_{1-N}, each having a score $s(T_i)$, the new score s' for the term T_i is obtained as follow:

$$s'(T_i) = s(T_i) + \alpha^{-1} \sum_{j \neq i} P(T_j|T_i) s(T_j)$$

where α is a constant in $[0 - 1]$ for controlling the re-ranking factor. α has been set experimentally to 0.8. $P(T_j|T_i)$ is the probability that the keyword T_j is chosen by the author when the keyword T_i has been selected. For obtaining these probabilities, we use statistics for the HAL[5] research archive. HAL contains approx. 139,000 full texts articles described by a rich set of metadata, often including author's keywords. We use the keywords appearing in English and in the Computer Science domain (a subset of 29,000 articles), corresponding to a total of 16,412 different keywords. No smoothing was used. The usage of open publication repository as a research resource is in its infancy and very promising.

6 Results

Our system was ranked first of the competition among 19 participants. Table 1 presents our official results (**P**recision, **R**ecall, **F**-score) for *combined* keywords and *reader* keywords, together with the scores of the systems ranked second (WINGNUS and KX FBK).

[4] http://wikipedia-miner.sourceforge.net

[5] HAL (Hyper Article en Ligne) is the French Institutional repository for research publications: http://hal.archives-ouvertes.fr/index.php?langue=en

Set	System	top 5	top 10	top 15
Comb.	HUMB	**P:**39.0 **R:**13.3 **F:**19.8	**F:**32.0 **R:**21.8 **F:**25.9	**P:**27.2 **R:**27.8 **F:**27.5
	WINGNUS	**P:**40.2 **R:**13.7 **F:**20.5	**P:**30.5 **R:**20.8 **F:**24.7	**P:**24.9 **R:**25.5 **F:**25.2
Reader	HUMB	**P:**30.4 **R:**12.6 **F:**17.8	**P:**24.8 **R:**20.6 **F:**22.5	**P:**21.2 **R:**26.4 **F:**23.5
	KX FBK	**P:**29.2 **R:**12.1 **F:**17.1	**P:**23.2 **R:**19.3 **F:**21.1	**P:**20.3 **R:**25.3 **F:**22.6

Table 1: Performance of our system (HUMB) and of the systems ranked second.

7 What did not work

The previously described features were selected because they all had a positive impact on the extraction accuracy based on our experiments on the task's collection. The following intuitively pertinent ideas appeared, however, to deteriorate or to be neutral for the results.

Noun phrase filtering We applied a filtering of noun phrases based on a POS tagging and extraction of all possible NP based on typical patterns. This filtering lowered both the recall and the precision (-7.6% for F-score at top 15).

Term variants We tried to apply a post-ranking by conflating term variants using FASTR[6], resulting in a disappointing -11.5% for the F-score.

Global keywordness We evaluated the keywordness using also the overall HAL keyword frequencies rather than only the training corpus. It had no impact on the results.

Language Model deviation We experimented the usage of HMM deviation using LingPipe[7] as alternative informativeness measure, resulting in -3.7% for the F-score at top 15.

Wikipedia term Relatedness Using Wikipedia Miner, we tried to apply as post-ranking a boosting of related terms, but saw no impact on the results.

8 Future work

We think that automatic key term extraction can be highly valuable for assisting self-archiving of research papers by authors in scholarly repositories such as arXiv or HAL. We plan to experiment keyword suggestions in HAL based on the present system. Many archived research papers are currently not associated with any keyword.

We also plan to adapt our module to a large collection of approx. 2.6 million patent documents in the context of CLEF IP 2010. This will be the opportunity to evaluate the relevance of the extracted key terms for large scale topic-based IR.

[6]http://perso.limsi.fr/jacquemi/FASTR

[7]http://alias-i.com/lingpipe

References

K. Ahmad and S. Collingham. 1996. Pointer project final report. Technical report, University of Surrey. http://www.computing.surrey.ac.uk/ai/pointer/report.

C.-C. Chang and C.-J. Lin. 2001. Libsvm: a library for support vector machines. Technical report.

P. Lopez and L. Romary. 2010. GRISP: A Massive Multilingual Terminological Database for Scientific and Technical Domains. In *Seventh international conference on Language Resources and Evaluation (LREC)*, Valletta, Malta.

P. Lopez. 2009. GROBID: Combining Automatic Bibliographic Data Recognition and Term Extraction for Scholarship Publications. In *Proceedings of ECDL 2009, 13th European Conference on Digital Library*, Corfu, Greece.

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

O. Medelyan. 2009. *Human-competitive automatic topic indexing*. Ph.D. thesis.

Y. Park, R.J. Byrd, and B.K. Boguraev. 2002. Automatic glossary extraction: beyond terminology identification. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.

F. Peng and A. McCallum. 2004. Accurate information extraction from research papers using conditional random fields. In *Proceedings of HLT-NAACL*, Boston, USA.

I.H. Witten and E. Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition edition.

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