DERIUNLP: A Context Based Approach to Automatic Keyphrase Extraction

Georgeta Bordea
Unit for Natural Language Processing
Digital Enterprise Research Institute
National University of Ireland, Galway
georgeta.bordea@deri.org

Paul Buitelaar
Unit for Natural Language Processing
Digital Enterprise Research Institute
National University of Ireland, Galway
paul.buitelaar@deri.org

Abstract

The DERI UNLP team participated in the SemEval 2010 Task #5 with an unsupervised system that automatically extracts keyphrases from scientific articles. Our approach does not only consider a general description of a term to select keyphrase candidates but also context information in the form of "skill types". Even though our system analyses only a limited set of candidates, it is still able to outperform baseline unsupervised and supervised approaches.

1 Introduction

Keyphrases provide users overwhelmed by the richness of information currently available with useful insight into document content but at the same time they are a valuable input for a variety of NLP applications such as summarization, clustering and searching. The SemEval 2010 competition included a task targeting the Automatic Keyphrase Extraction from Scientific Articles (Kim et al., 2010). Given a set of scientific articles participants are required to assign to each document keyphrases extracted from text.

We participated in this task with an unsupervised approach for keyphrase extraction that does not only consider a general description of a term to select candidates but also takes into consideration context information. The larger context of our work is the extraction of expertise topics for Expertise Mining (Bordea, 2010).

Expertise Mining is the task of automatically extracting expertise topics and expertise profiles from a collection of documents. Even though the Expertise Mining task and the Keyphrase Extraction task are essentially different, it is important to assess the keyphraseness of extracted expertise topics, i.e., their ability to represent the content of a document. Here we will report only relevant findings for the Keyphrase Extraction task, focusing on the overlapping aspects of the two aforementioned tasks.

After giving an overview of related work in section 2 we introduce skill types and present our candidate selection method in section 3. Section 4 describes the features used for ranking and filtering the candidate keyphrases and Section 5 presents our results before we conclude in Section 6.

2 Related Work

The current methods for keyphrase extraction can be categorized in supervised and unsupervised approaches. Typically any keyphrase extraction system works in two stages. In the first stage a general set of candidates is selected by extracting the tokens of a text. In the second stage unsupervised approaches combine a set of features in a rank to select the most important keyphrases and supervised approaches use a training corpus to learn a keyphrase extraction model.

Mihalcea and Tarau (2004) propose an unsupervised approach that considers single tokens as vertices of a graph and co-occurrence relations between tokens as edges. Candidates are ranked using PageRank and adjacent keywords are merged into keyphrases in a post-processing step. The frequency of noun phrase heads is exploited by Barker and Cornacchia (2000), using noun phrases as candidates and ranking them based on term frequency and term length.

Kea is a supervised system that uses all n-grams of a certain length, a Naive Bayes classifier and tf-idf and position features (Frank et al., 1999). Turney (2000) introduces Extractor, a supervised system that selects stems and stemmed n-grams as candidates and tunes its parameters (mainly related to frequency, position, length) with a genetic algorithm. Hulth (2004) experiments with three types of candidate terms (i.e., n-grams, noun phrase chunks and part-of-speech tagged words

Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 146–149,
Uppsala, Sweden, 15-16 July 2010. ©2010 Association for Computational Linguistics

that match a set of patterns) and constructs classifiers by rule induction using features such as term frequency, collection frequency, relative position and PoS tags.

The candidate selection method is the main difference between our approach and previous work. We did not use only a general description of a term to select candidates, but we also took into consideration context information.

3 The Skill Types Candidate Selection Method

Skill types are important domain words that are general enough to be used in different subfields and that reflect theoretical or practical expertise. Consider for instance the following extracts from scientific articles:

*...**analysis** of historical trends...*
*...duplicate photo detection **algorithm** ...*
*...**approach** for data assimilation...*
*...**methodology** for reservoir characterization...*

In all four examples the expertise topic (e.g., "historical trends", "duplicate photo detection algorithm", "data assimilation", "reservoir characterization") is introduced by a skill type (e.g., "analysis", "algorithm", "approach", "methodology"). Some of these skill types are valid for any scientific area (e.g. "approach", "method", "analysis", "solution") but we can also identify domain specific skill types, e.g., for computer science "implementation", "algorithm", "development", "framework", for physics "proof", "principles", "explanation" and for chemistry "law", "composition", "mechanism", "reaction", "structure".

Our system is based on the GATE natural language processing framework (Cunningham et al., 2002) and it uses the ANNIE IE system included in the standard GATE distribution for text tokenization, sentence splitting and part-of-speech tagging. The GATE processing pipeline is depicted in Figure 1, where the light grey boxes embody components available as part of the GATE framework whereas the dark grey boxes represent components implemented as part of our system.

We manually extract a set of 81 single word skill types for the Computer Science field by analysing word frequencies for topics from the ACM classification system[1]. The skill types that appear most

[1]ACM classification system: `http://www.acm.org/about/class/`

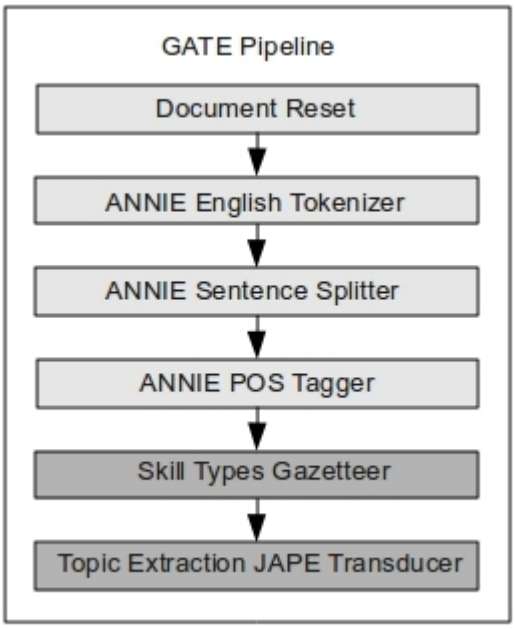

Figure 1: GATE Processing Pipeline

frequently in keyphrases given in the training set are "system", "model" and "information". The Skill Types Gazetteer adds annotations for skill types and then the JAPE Transducer uses regular expressions to annotate candidates.

We rely on a syntactic description of a term to discover candidate keyphrases that appear in the right context of a skill type or that include a skill type. The syntactic pattern for a term is defined by a sequence of part-of-speech tags, mainly a noun phrase. We consider that a noun phrase is a head noun accompanied by a set of modifiers (i.e nouns, adjectives) that includes proper nouns, cardinal numbers (e.g., "P2P systems") and gerunds (e.g., "ontology mapping", "data mining"). Terms that contain the preposition "of" (e.g., "quality of service") or the conjunction "and" (e.g., "search and rescue") were also allowed.

4 Ranking and Filtering

For the ranking stage we use several features already proposed in the literature such as length of a keyphrase, tf-idf and position. We also take into consideration the collection frequency in the context of a skill type.

Ranking. Longer candidates in terms of number of words are ranked higher, because they are more descriptive. Keyphrases that appear more frequently with a skill type in the collection of documents are also ranked higher. Therefore we define the rank for a topic as:

Method	5P	5R	5F	10P	10R	10F	15P	15R	15F
TF-IDF	22	7.5	11.19	17.7	12.07	14.35	14.93	15.28	15.1
NB	21.4	7.3	10.89	17.3	11.8	14.03	14.53	14.87	14.7
ME	21.4	7.3	10.89	17.3	11.8	14.03	14.53	14.87	14.7
DERIUNLP	**27.4**	**9.35**	**13.94**	**23**	**15.69**	**18.65**	**22**	**22.51**	**22.25**
DUB	15.83	5.13	7.75	13.40	8.68	10.54	13.33	12.96	13.14

Table 1: Baseline and DERIUNLP Performance aver Combined Keywords

System	5P	5R	5F	10P	10R	10F	15P	15R	15F
Best	39.0	13.3	19.8	32.0	21.8	26.0	27.2	27.8	27.5
Average	29.6	10.1	15	26.1	17.8	21.2	21.9	22.4	22.2
Worst	9.4	3.2	4.8	5.9	4.0	4.8	5.3	5.4	5.3
DERIUNLP	27.4	9.4	13.9	23.0	15.7	18.7	22.0	22.5	22.3

Table 2: Performance over Combined Keywords

$$R_{i,j} = Tn_i * Fn_i * tfidf_{i,j}$$

Where R_i is the rank for the candidate i and the document j, Tn_i is the normalized number of tokens (number of tokens divided by the maximum number of tokens for a keyphrase), Fn_i is the normalized collection frequency of the candidate in the context of a skill type (collection frequency divided by the maximum collection frequency), and $tfidf_i$ is the TF-IDF for candidate i and topic j (computed based on extracted topics not based on all words).

Filtering. Several approaches (Paukkeri et al., 2008; Tomokiyo and Hurst, 2003) use a reference corpus for keyphrase extraction. We decided to use the documents available on the Web as a reference corpus, therefore we use an external web search engine to filter out the candidates that are too general from the final result set. If a candidate has more than 10^9 hits on the web it is too general to be included in the final result set. A lot of noise is introduced by general combination of words that could appear in any document. We remove candidates longer than eight words and we ignore keyphrases that have one letter words or that include non-alphanumerical characters.

Acronyms. Acronyms usually replace long or frequently referenced terms. Results are improved by analysing acronyms (Krulwich and Burkey, 1996) because most of the times the expanded acronym is reported as a keyphrase, not the acronym and because our rank is sensitive to the number of words in a keyphrase. We consider the length of an acronym to be the same as the length of its expansion and we report only the expansion as a keyphrase.

Position. The candidates that appear in the title or the introduction of a document are more likely to be relevant for the document. We divide each document in 10 sections relative to document size and we increase the ranks for keyphrases first mentioned in one of these sections (200% increase for the first section, 100% increase for the second section and 25% for the third section). Candidates with a first appearance in the last section of a document are penalised by 25%.

5 Evaluation

The SemEval task organizers provided two sets of scientific articles, a set of 144 documents for training and a set of 100 documents for testing. No information was provided about the scientific domain of the articles but at least some of them are from Computer Science. The average length of the articles is between 6 and 8 pages including tables and pictures. Three sets of answers were provided: author-assigned keyphrases, reader-assigned keyphrases and combined keyphrases (combination of the first two sets). The participants were asked to assign a number of exactly 15 keyphrases per document.

All reader-assigned keyphrases are extracted from the papers, whereas some of the author-assigned keyphrases do not occur explicitly in the text. Two alternations of keyphrase are accepted: A of B / B A and A's B. In case that the semantics changes due to the alternation, the alternation is not included in the answer set. The traditional evaluation metric was followed, matching the extracted keyphrases with the keyphrases in the answer sets and calculating precision, recall and F-score. In both tables the column labels start with a number which stands for the top 5, 10 or 15 candidates. The characters P, R, F mean micro-averaged precision, recall and F-scores. For baselines, 1, 2, 3 grams were used as candidates and TF-IDF as features.

In Table 1 the keyphrases extracted by our system are compared with keyphrases extracted by

an unsupervised method that ranks the candidates based on TF-IDF scores and two supervised methods using Naive Bayes (NB) and maximum entropy(ME) in WEKA[2]. Our performance is well above the baseline in all cases.

To show the contribution of skill types we included the results for a baseline version of our system (DUB) that does not rank the candidates using the normalized collection frequency in the context of a skill type Fn_i but the overall collection frequency (i.e., the number of occurrences of a keyphrase in the corpus). The significantly increased results compared to our baseline version show the effectiveness of skill types for keyphrase candidate ranking.

Table 2 presents our results in comparison with results of other participants. Even though our system considers in the first stage a significantly limited set of candidates the results are very close to the average results of other participants. Our system performed 8th best out of 19 participants for top 15 keyphrases, 10th best for top 10 keyphrases and 13th best for top 5 keyphrases, which indicates that our approach could be improved by using a more sophisticated ranking method.

6 Conclusions

In this paper we have reported the performance of an unsupervised approach for keyphrase extraction that does not only consider a general description of a term to select keyphrase candidates but also takes into consideration context information. The method proposed here uses term extraction techniques (the syntactic description of a term), classical keyword extraction techniques(TF-IDF, length, position) and contextual evidence (skill types).

We argued that so called "skill types" (e.g., "methods", "approach", "analysis") are a useful instrument for selecting keyphrases from a document. Another novel aspect of this approach is using the collection of documents available on the Web (i.e., number of hits for a keyphrase) instead of a reference corpus. It would be interesting to evaluate the individual contributions of skill types for Keyphrase Extraction by adding them as a feature in a classical system like KEA.

Future work will include an algorithm for automatic extraction of skill types for a domain and an analysis of the performance of each skill type.

[2]WEKA:http://www.cs.waikato.ac.nz/ml/weka/

7 Aknowledgements

This work is supported by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2).

References

Ken Barker and Nadia Cornacchia. 2000. Using Noun Phrase Heads to Extract Document Keyphrases. In *Canadian Conference on AI*, pages 40–52. Springer.

Georgeta Bordea. 2010. Concept Extraction Applied to the Task of Expert Finding. In *Extended Semantic Web Conference 2010, PhD Symposium*. Springer.

H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*.

Eibe Frank, Gordon W Paynter, Ian H Witten, Carl Gutwin, and Craig G Nevill-Manning. 1999. Domain-Specific Keyphrase Extraction. In *Proceedings of the 16th International Joint Conference on Aritfiicial Intelligence*, pages 668–673.

Anette Hulth. 2004. Enhancing Linguistically Oriented Automatic Keyword Extraction. In *Proceedings of HLT/NAACL: Short Papers*, pages 17–20.

Su Nam Kim, Alyona Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles. In *Proceedings of the ACL 2010 Workshop on Evaluation Exercises on Semantic Evaluation (SemEval 2010)*.

Bruce Krulwich and Chad Burkey. 1996. Learning user information interests through extraction of semantically significant phrases. In *Proc. AAAI Spring Symp. Machine Learning in Information Access*, Menlo Park, Calif. Amer. Assoc. for Artificial Intelligence.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 404–411.

Mari-Sanna Paukkeri, Ilari T. Nieminen, Polla Matti, and Timo Honkela. 2008. A Language-Independent Approach to Keyphrase Extraction and Evaluation. In *Coling 2008 Posters*, number August, pages 83–86.

Takashi Tomokiyo and Matthew Hurst. 2003. A Language Model Approach to Keyphrase Extraction. In *Proceedings of the ACL 2003 work- shop on Multiword expressions*, pages 33–40.

Peter D Turney. 2000. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2:303–336.