LIPN at SemEval-2017 Task 10: Filtering Candidate Keyphrases from Scientific Publications with Part-of-Speech Tag Sequences to Train a Sequence Labeling Model

Simon David Hernandez, Davide Buscaldi, Thierry Charnois
Laboratoire d'Informatique de Paris Nord, CNRS (UMR 7030)
Université Paris 13, Sorbonne Paris Cité, Villetaneuse, France
{hernandez-perez,buscaldi,thierry.charnois}@lipn.univ-paris13.fr

Abstract

This paper describes the system used by the team LIPN in SemEval 2017 Task 10: Extracting Keyphrases and Relations from Scientific Publications. The team participated in Scenario 1, that includes three subtasks, Identification of keyphrases (Subtask A), Classification of identified keyphrases (Subtask B) and Extraction of relationships between two identified keyphrases (Subtask C). The presented system was mainly focused on the use of part-of-speech tag sequences to filter candidate keyphrases for Subtask A. Subtasks A and B were addressed as a sequence labeling problem using Conditional Random Fields (CRFs) and even though Subtask C was out of the scope of this approach, one rule was included to identify synonyms.

1 Introduction

Identifying candidate keyphrases in texts is commonly a first step in systems for keyphrase extraction (Hasan and Ng, 2014; Haddoud et al., 2015), this can be done by filtering words or phrases from documents using heuristics to determine which can be candidate keyphrases.

The system uses *sequences of part-of-speech tags* (*PoS sequences*) as patterns to filter candidate keyphrases. These candidates are used to train two Conditional Random Field (CRF) models, one for keyphrase identification and other for keyphrase classification. CRF was trained with orthographic features, additionally to features from WordNet and titles from academic papers. The PoS sequences were extracted from the annotated keyphrases in the corpus provided for the task (Augenstein et al., 2017).

The PoS sequences used in this system are described in Section 2, there is an explanation of how they were used and how they were extracted from the training data. In Section 3 is detailed how CRF was trained with the candidate keyphrases and in Section 4 the features are described. In Section 5 there is an explanation of how CRF was applied to identify and classify keyphrases. Section 6 shows the post-processing steps and Section 7 introduces some experiments.

2 PoS sequences

In this paper, we use the term *PoS sequences* to refer to *sequences of part-of-speech tags*. PoS sequences are used in automatic keyphrase extraction as features (Kim and Kan, 2009; Hasan and Ng, 2014) or to filter candidate keyphrases (Kim and Kan, 2009; Haddoud et al., 2015; Hasan and Ng, 2014), for example, with small sets of patterns matching all noun phrases and prepositional phrases, avoiding patterns that increase error, like sequences containing adverbs (Kim and Kan, 2009).

In this system, PoS sequences are used only to filter candidate keyphrases. From the annotated keyphrases in the training data, were extracted 1445 different PoS sequences [1], Table 1 shows an example of PoS sequences, sorted by number of occurrences.

Each extracted PoS sequence was used as a pattern to filter candidate keyphrases in the training, development and test corpus, instead of generalize a smaller set of patterns as is proposed in other approaches.

[1] The full list of extracted POS sequences is available in https://github.com/snovd/corpus-data/blob/master/SemEval2017Task10/POSsequences.txt

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 995–999,
Vancouver, Canada, August 3 - 4, 2017.

Occurrences	POS sequence
1333	NN
559	NN NN
414	JJ NN
301	NN NNS
293	NNS
289	JJ NNS
⋮	⋮
61	VBG
54	NN NN NNS
52	JJ JJ NN
51	JJ
41	VBG NN
⋮	⋮

Table 1: Example of POS sequences extracted from the training data, ordered by number of occurrences.

2.1 Filtering candidate keyphrases

Each PoS sequence is compared with the part-of-speech[2] of a text, all the sequences of tokens matching the pattern are selected as candidate keyphrases.

> provides/VBZ an/DT approach/NN to/TO circumvent/VB the/DT sign/NN problem/NN in/IN numerical/JJ simulations/NNS

Figure 1: Extract from the development data[3].

For example, the extract of text in Figure 1 has the following annotations, *"sign problem"* is a keyphrase of type TASK and *"numerical simulations"* is part of a larger keyphrase of type PROCESS. From the same text, two sets of candidate keyphrases are shown in Table 2, the first set is obtained by matching all the PoS sequences and the second by matching the PoS sequences with at least 14 occurrences.

If we were using ngrams to propose candidate keyphrases, in same example, we get 45 different candidates, with ngrams from 1-grams to 5-grams, so there is a significant reduction of extracted phrases. Also, note that there is a reduction of candidate keyphrases between the two sets in Table 2 without excluding the annotated keyphrases, also the token *"provides"* is missing. We took advantage of this observation to improve the precision, see Section 7.

[2]Getting the PoS with TreebankWordTokenizer and PerceptronTagger in NLTK

[3]File S0003491613001516.txt

Occurrences of PoS sequence	Extracted phrases
≥ 1	problem in numerical simulations the sign problem circumvent the sign problem sign sign problem in numerical simulations provides an approach **numerical simulations** the sign approach circumvent the sign approach to circumvent an approach problem in numerical problem an simulations problem in numerical the **sign problem**
≥ 14	problem in numerical simulations sign **sign problem** approach **numerical simulations** problem simulations numerical

Table 2: Two sets of candidate keyphrases. Generated with the PoS sequences filtered by the number of occurrences.

2.2 Keyphrases and Non-keyphrases

We extracted all the possible candidate keyphrases from the training corpus, using all the PoS sequences described before. An extracted candidate is labeled as KEYPHRASE if it is annotated as keyphrase in the training corpus, on the contrary it is labeled as NON-KEYPHRASE, like in a binary classification problem (Frank et al., 1999).

3 Training CRF

Using Conditional Random Fields (CRFs) to address Automatic Keyword Extraction as a se-

quence labeling problem has already been proposed (Bhaskar et al., 2012; Zhang, 2008; Augenstein et al., 2017).

We trained CRF [4] only with the candidate keyphrases, each one as a separated input, using BIO encoding[5] and the labels KEYPHRASE and NON-KEYPHRASE for Subtask A, like in the examples in Figures 2 and 3. Similarly, a second CRF model was trained for Subtask B, with labels TASK, PROCESS and MATERIAL.

the/O sign/B problem/I in/O

Figure 2: Example of KEYPHRASE/TASK 'sign problem' in BIO encoding.

'the/O sign/B problem/I in/I numerical/I simulations/I of/O'

Figure 3: Example of NON-KEYPHRASE 'sign problem in numerical simulations' in BIO encoding.

Note that in the sets in Table 2 there are repetitions of tokens in several candidate keyphrases. For example, *"sign problem"* is an annotated keyphrase, so it is labeled as KEYPHRASE/TASK, in contrast with *"circumvent the sign problem"* and *"sign problem in numerical simulations"* which are labeled as NON-KEYPHRASE, ignoring completely that these phrases contain a keyphrase. Also, text that doesn't match a PoS sequence is not used to train the model.

4 Features

For identification of keyphrases (Subtask A) and classification of identified keyphrases (Subtask B), we trained two different CRF models with the same candidate keyphrases, labeled differently depending on the subtask. Subtask B uses the same features that Subtask A, in addition to features from WordNet.

All the features were generated for each token in a given candidate, including the tokens that surround the start and end of the phrase, as shown in the examples of Figures 2 and 3. Text that is not present in the candidate keyphrases is ignored with the exception of these two context tokens as features.

4.1 Features - Subtask A

To train CRF for Subtask A, we used the features suggested in the documentation of python-crfsuite for the task of named entity recognition, we didn't make a deep exploration of them. Those features are the token in lowercase, its part-of-speech, the first two letters of the part-of-speech, the suffixes of one and two characters, and three binary features, which value depends on the letter case of the token, these are uppercase, lowercase or title case, also are included two tokens of context (previous, next) in lowercase. Finally, an indicator is added if the token is at the beginning or the end of the whole text.

4.1.1 Titles from academic papers

Information from titles has been useful in keyphrase extraction (Hasan and Ng, 2014; Grineva et al., 2009), so we generated a database with bigrams, trigrams and the part-of-speech of the trigrams, extracted from titles from academic papers[6]. Only titles in English were included[7].

We added four binary features for each token in a candidate keyphrase, the value depends on whether ngrams formed with its context exist or not in the database. For example, the token *'sign'* in Figure 1 forms the ngrams, *'the sign'*, *'sign problem'*, *'the sign problem'* and *'DT NN NN'*.

4.2 Features - Subtask B

We used binary features with information from WordNet 3.0, these are included only when the lemmatized token [8] has *noun synsets*. The first feature is True only if the synsets of the lemmatized token have holonyms, a second feature depends on whether it has derivationally related forms.

We also included a fixed set of synsets as binary features[9], which are the more probable synsets

[4]We used python-crfsuite with the default parameters for Named Entity Recognition, 'c1': 1.0, 'c2': 1e-3, 'max_iterations': 50, 'feature.possible_transitions': True, https://github.com/scrapinghub/python-crfsuite

[5]Indicating the (B)eginning of the phrase, (I)nside of the phrase or (O)ther.

[6]Microsoft Academic Graph, version 2016/02/05 https://academicgraphwe.blob.core.windows.net/graph-2016-02-05/index.html

[7]Were separated with guess_language https://pypi.python.org/pypi/guess_language-spirit

[8]Lemmatization with WordNetLemmatizer

[9]List of synsets used as binary features. https://github.com/snovd/corpus-data/blob/master/SemEval2017Task10/SynsetsRelatedToTrainingData.txt

from the annotations in the training corpus. To obtain the set, we merged the twenty[10] more probable synsets by label (PROCESS, MATERIAL and TASK). These features are True for a token if they are present in the hypernyms of the noun synsets.

5 Identifying and labeling keyphrases

First, CRF was trained as described in Section 3, then we extracted the candidate keyphrases from the development/test data with the PoS sequences having at least 14 occurrences in the training data with the method explained in Section 2. Then we excluded the candidates of one token if they exist in an exclusion list (described in Subsection 5.1).

Then CRF was applied to each candidate keyphrase with the features for Subtask A, if all the tokens in the candidate were labeled as KEYPHRASE, then the entire candidate was labeled as KEYPHRASE.

CRF was applied again to all the resulting keyphrases from the last steps, but this time with the model trained with the Subtask B features. Similarly, if the tokens in the keyphrase were labeled with the same type, then the keyphrase is labeled entirely with the corresponding type, PROCESS, MATERIAL or TASK. If the keyphrase was not labeled equally, it was marked as PROCESS by default.

5.1 Exclusion list

This list was generated from the training corpus to exclude very common tokens. It was generated by filtering the inverse document frequency (idf) of each token with a threshold. First, we calculated the idf for all the tokens in the papers from the training corpus. The threshold is the mean of the idfs minus four times the standard deviation. One token is added to the exclusion list only if its idf is lesser or equal than the threshold.

6 Post-processing

For the case of overlapping, as it is shown in Table 5, when a full keyphrase is contained inside other keyphrase, the largest keyphrase is chosen.

Finally, we included a simple rule to relate synonyms. By observation of the training data, we noticed that two keyphrases are marked as synonyms, if one is followed by another inside of parenthesis, been the second an acronym of the first.

[10]This number was chosen arbitrarily

7 Experiments

In Figures 4 (Precision), 5 (Recall) and 6 (F_1) are shown the results of different experiments for Subtask A. In these experiments we tested the effect of removing the least occurring PoS sequences in the training corpus to filter the candidate keyphrases in the development corpus. *"Candidate keyphrases + CRF + Titles"* represents the experiments of the system with all the features as described previously. *"Candidate keyphrases + CRF"* represents the experiments of the system without using the database of titles as features (Subsection 4.1.1). *"CRF"* and *"CRF + Titles"* are the results of applying CRF with the same features and without filtering candidate keyphrases. *"Candidate keyphrases"* is our baseline, these are the results of using candidate keyphrases directly as keyphrases.

As can be observed in Figure 6, the best F_1 score is reached when the candidate keyphrases from the development corpus were filtered with all the PoS sequences with an occurrence of at least 14 times in the training corpus, like in the example of Table 2 and as described in Section 5. In that case, the proposed system has a better result in F_1 score than *"CRF + Titles"*.

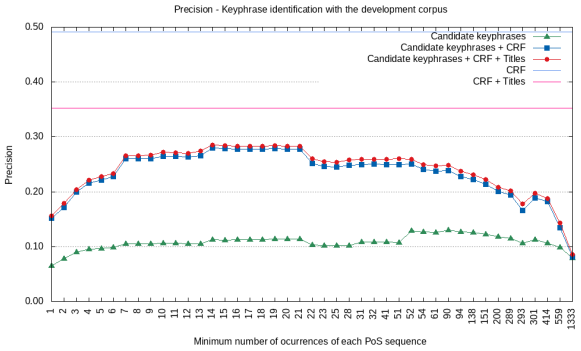

Figure 4: Precision: Experiments for Subtask A with the development corpus.

8 Results

Our final results are shown in Table 3, we ranked 11th in Scenario 1, 10th in Subtask A and 11th in Subtask B. We obtained our best performance in Subtask A, which is the main target of this work.

9 Conclusion

We tested the use of PoS sequences extracted from the training data to filter candidate keyphrases, in-

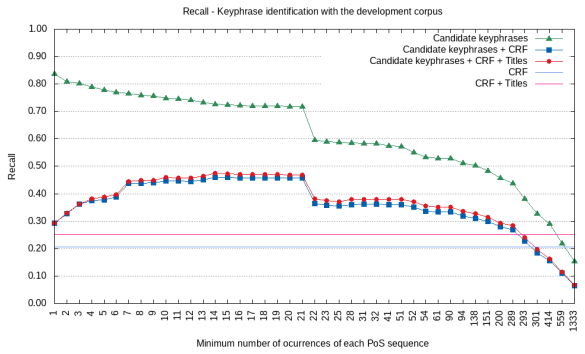

Figure 5: Recall: Experiments for Subtask A with the development corpus.

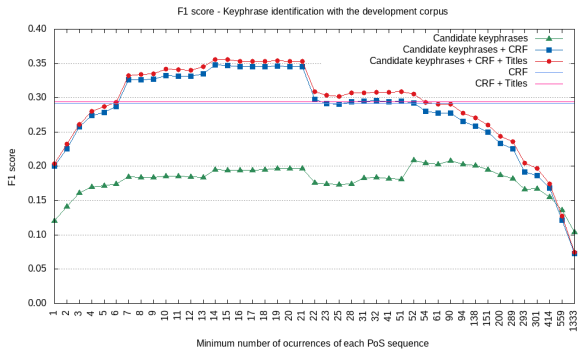

Figure 6: F_1 score: Experiments for Subtask A with the development corpus.

stead of filtering with a fixed set of patterns to match noun phrases or prepositional phrases as proposed in other approaches. Our experiments show that filtering candidate keyphrases to train CRF with this method helps to improve the results for Automatic Keyphrase Extraction by increasing the Recall, with the disadvantage of lost of Precision.

Subtask	P	R	F_1
Scenario 1	0.17	0.25	0.21
Subtask A	**0.31**	**0.49**	**0.38**
Subtask A+B	0.17	0.27	0.21
Subtask C	0.33	0.02	0.05

Table 3: Final results for team LIPN in SemEval 2017 Task 10

References

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. Semeval 2017 task 10: Scienceie - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, pages 537–546. http://www.aclweb.org/anthology/S17-2091.

Pinaki Bhaskar, Kishorjit Nongmeikapam, and Sivaji Bandyopadhyay. 2012. Keyphrase extraction in scientific articles: A supervised approach. In *Proceedings of COLING 2012: Demonstration Papers*. The COLING 2012 Organizing Committee, Mumbai, India, pages 17–24. http://www.aclweb.org/anthology/C12-3003.

Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, IJCAI '99, pages 668–673. http://dl.acm.org/citation.cfm?id=646307.687591.

Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. 2009. Extracting key terms from noisy and multitheme documents. In *Proceedings of the 18th International Conference on World Wide Web*. ACM, New York, NY, USA, WWW '09, pages 661–670. https://doi.org/10.1145/1526709.1526798.

Mounia Haddoud, Aïcha Mokhtari, Thierry Lecroq, and Saïd Abdeddaïm. 2015. Accurate Keyphrase Extraction from Scientific Papers by Mining Linguistic Information. *Proceedings of the First Workshop on Mining Scientific Papers: Computational Linguistics and Bibliometrics co-located with 15th International Society of Scientometrics and Informetrics Conference (ISSI 2015) Istanbul, Turkey, June 29, 2015.* 1384:12–17. http://ceur-ws.org/Vol-1384/paper2.pdf.

Kazi Saidul Hasan and Vincent Ng. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 1262–1273. http://www.aclweb.org/anthology/P/P14/P14-1119.

Su Nam Kim and Min-Yen Kan. 2009. Re-examining automatic keyphrase extraction approaches in scientific articles. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*. Association for Computational Linguistics, Singapore, pages 9–16. http://www.aclweb.org/anthology/W/W09/W09-2902.

Chengzhi Zhang. 2008. Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems* 4(3):1169–1180.