SZTE-NLP at SemEval-2017 Task 10: A High Precision Sequence Model for Keyphrase Extraction Utilizing Sparse Coding for Feature Generation

Gábor Berend
Department of Informatics, University of Szeged
Árpád tér 2, H6720 Szeged, Hungary
berendg@inf.u-szeged.hu

Abstract

In this paper we introduce our system participating at the 2017 SemEval shared task on keyphrase extraction from scientific documents. We aimed at the creation of a keyphrase extraction approach which relies on as little external resources as possible. Without applying any hand-crafted external resources, and only utilizing a transformed version of word embeddings trained at Wikipedia, our proposed system manages to perform among the best participating systems in terms of precision.

1 Introduction

The sheer amount of scientific publications makes intelligent processing of papers increasingly important. Automated keyphrase extraction techniques can mitigate the severe difficulties arising when navigating in massive document collections. Hence, extracting keyphrases from scientific literature has generated substantial academic interest over the past years (Witten et al., 1999; Hulth, 2003; Kim et al., 2010; Berend, 2016a).

Continuous word representations such as `word2vec` (Mikolov et al., 2013) has gained increasing popularity recently. These representations assign some semantically meaningful low dimensional vector $\mathbf{w}_i$ to the vocabulary entries of large text corpora.

We demonstrated previously (Berend, 2016b) that useful features can be derived for various sequence labeling tasks by performing a sparse decomposition of the word embedding matrix. In this paper, we investigate the generalization properties of our proposed approach for the task of keyphrase extraction.

2 Sequence labeling framework

Our sequence labeling framework builds on top of our previous work which aimed at multiple different sequence labeling tasks, i.e. part-of-speech tagging and named entity recognition.

2.1 Feature representation

Each token in a sequence is described by a set of feature values and those of its direct neighbors in our model. We relied on multiple sources for deriving features, i.e.

- sparse coding of dense word embeddings,
- Brown clustering of words,
- word identity features and
- orthographic characteristics.

2.1.1 Sparse coding derived features

The main source of features was sparse coding performed on continuous word embeddings. We demonstrated in (Berend, 2016b) that sequence labeling tasks can largely benefit from the sparse decomposition of dense word embedding matrices. That is, given a word embedding matrix $W \in \mathbb{R}^{d \times |V|}$ – with its columns containing the d dimensional dense word embeddings – we seek for its decomposition into a product of $D \in \mathbb{R}^{d \times K}$ and $\alpha \in \mathbb{R}^{K \times |V|}$ – containing sparse linear combination coefficients for each of the word embeddings – such that $\|W - D\alpha\|_F^2 + \lambda\|\alpha\|_1$ gets minimized.

Features for some word $\boldsymbol{w}_i$ are then determined based on its corresponding vector $\boldsymbol{\alpha}_i$ by taking the signs and indices of its non-zero coefficients, i.e.

$$f(\mathbf{w}_i) = \{sign(\boldsymbol{\alpha}_i[j])j \mid \boldsymbol{\alpha}_i[j] \neq 0\},$$

where $\boldsymbol{\alpha}_i[j]$ denotes the j^{th} coefficient in $\boldsymbol{\alpha}_i$.

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 990–994,
Vancouver, Canada, August 3 - 4, 2017.

As we observed a consistent benefit of using polyglot (Al-Rfou et al., 2013) embeddings previously, we now also rely on those embeddings for keyphrase extraction.

2.1.2 Brown clustering

Brown clustering (Brown et al., 1992) defines a hierarchical clustering over words and cluster supersets can be easily turned into features. We used the commonly employed approach of deriving features from the length-p ($p \in \{4, 6, 10, 20\}$) prefixes of Brown cluster identifiers as it was done previously by Ratinov and Roth (2009); Turian et al. (2010) as well.

We used the implementation of Liang (2005) for determining 1024 Brown clusters[1] based on the same Wikipedia dump which was used upon the training of the freely available `polyglot` word embeddings[2] that we relied on for performing sparse decomposition.

2.1.3 Orthographic features

Orthographic clues can vastly help identifying keyphrases in scientific publications. For this reason the below listed indicator features get determined for some word w:

- $isNumber(w)$
- $isTitleCase(w)$
- $isNonAlnum(w)$
- $containsNonAlnum(w)$
- $prefix(w, i)$ for $1 \leq i \leq 4$
- $suffix(w, i)$ for $1 \leq i \leq 4$

2.2 Training the model

Features described in Section 2.1 were utilized in linear chain CRFs (Lafferty et al., 2001) relying on the CRFsuite (Okazaki, 2007) implementation. CRFSuite was applied with its default regularization parameters, i.e. 1.0 and 0.001 for ℓ_1 and ℓ_2 regularization, respectively.

The shared task also required the identification of keyphrase types beyond merely finding the keyphrases within the text. We handled the fact that keyphrase scopes of different keyphrase types could overlap by training a separate CRF model for each keyphrase type and merging the predictions of the different models in a post-processing step. The models we trained employ the 5-class BIOES-augmented tagging scheme for the labels.

[1] https://github.com/percyliang/brown-cluster

[2] https://sites.google.com/site/rmyeid/projects/polyglot

	Sentence	Word form	Token
Train	35.10%	77.77%	94.59%
Dev	36.19%	86.77%	94.84%
Test	31.84%	83.48%	94.49%

(a) Overall word representation coverages.

	Material	Process	Task
Train	85.03%	91.65%	93.55%
Dev	82.60%	92.05%	96.21%
Test	80.35%	88.84%	93.14%

(b) Per-category token-level coverage breakdown.

Table 1: Coverages of the word embeddings.

3 Experiments

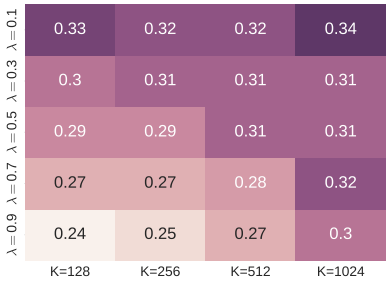
In this section we report our evaluations on the SemEval-2017 Task 10 dataset which consists of 350 training, 50 development and 100 test text passages, respectively. Each text passage originates from either Computer Science, Material Sciences or Physics publications and the task was to identify and classify keyphrases into the types of *Material*, *Process* and *Task*.

The shared task included both a keyphrase type insensitive (Subtask A) and sensitive (Subtask B) evaluation. Further details about the dataset and the description of the keyphrase types can be accessed in (Augenstein et al., 2017).

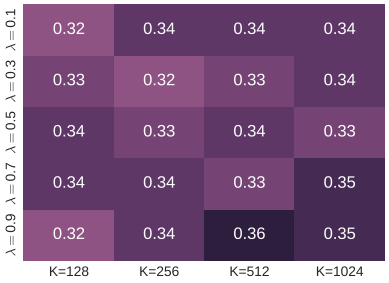
The only preprocessing we performed on the shared task data was sentence splitting and tokenization of input sentences. These steps were executed relying on `spacy`[3]. In order the sparse word embedding and Brown clustering-based features to work effectively, it is important that the a substantial amount of tokens from the shared task data have word representation determined for, i.e. the coverage of the word representations is satisfactory.

Table 1 includes the coverage of the word representations for the training, development and test sets. Table 1a contains the proportion of sentences with all words having a word representation determined for, alongside with the same values for

[3] https://spacy.io



(a) Excluding word identity features



(b) Including word identity features

Figure 1: Micro-averaged F-scores for Subtask B as a function of varying λ and K parameters for sparse coding without Brown clustering-based and orthographic features being used.

	Precision	Recall	F-score
Subtask A	0.51	0.27	0.35
Subtask B avg.	0.40	0.21	0.28
Material	0.46	0.27	0.34
Process	0.39	0.19	0.26
Task	0.09	0.05	0.06

(a) Excluding word identity features.

	Precision	Recall	F-score
Subtask A	0.51	0.30	0.38
Subtask B avg.	0.39	0.23	0.29
Material	0.43	0.29	0.35
Process	0.38	0.20	0.27
Task	0.14	0.05	0.07

(b) Including word identity features.

Table 2: Results of the official submission on the test data with $K = 128, \lambda = 0.9$.

word forms and tokens. Table 1b provides a more detailed breakdown of the coverages of word representations for the different keyphrase types also.

As subsequent results illustrate, higher word coverage for a certain type of keyphrase does not necessarily imply better performance on that type as e.g. *Task*-type keyphrases have the highest token coverage, nevertheless, scores are the lowest on that particular type (cf. Table 4).

3.1 Results on development data

Figure 1 illustrates the effect of varying the K and λ hyperparameters of sparse coding when not relying on orthographic or Brown clustering derived features. Figure 1b illustrates the effect of adding word identity features to the sparse coding derived ones, which suggests that the choice of $K = 1024$ seems to a reasonable choice for sparse coding since for that value of K, adding word identity features over the sparse coding derived ones yields marginal (or no) improvements. Inspecting Figure 1a also reveals that setting the regualrization parameter λ too high hurts performance.

	Precision	Recall	F-score
Subtask A	0.49	0.25	0.33
Subtask B avg.	0.37	0.19	0.25
Material	0.42	0.26	0.32
Process	0.36	0.15	0.21
Task	0.13	0.05	0.07

Table 3: Results on the test set with all features used **except for** the sparse coding-derived ones.

Subsequently, we investigate how does adding orthographic and Brown clustering-derived features affect results for two extremely different hyperparameter combinations of sparse coding, i.e. $K = 128, \lambda = 0.9$ and $K = 1024, \lambda = 0.1$. These results are presented in Table 4a-4d. Table 4 reveals that when orthographic and/or Brown clustering-based features are used in conjunction with the sparse coding derived ones, results become more stable, i.e. they are much less affected by the choices of the K and λ. Simultaneously, the importance of word identity features diminishes once orthographic and/or Brown clustering-related ones get involved in the model. This effect is more pronounced when adding orthographic features.

Interestingly, when both orthographic and Brown clustering related features are employed, results become better for small values of K, however, this was not the case without the application of these additional feature classes.

	Sparse coding only			+Brown			+Orthograpy			+Brown+Orthography		
	P	R	F	P	R	F	P	R	F	P	R	F
Subtask A	0.69	0.18	0.28	0.64	0.25	0.36	0.63	0.32	0.42	0.61	0.34	0.44
Subtask B avg.	0.59	0.15	0.24	0.56	0.22	0.31	0.53	0.27	0.36	0.54	0.30	0.39
Material	0.63	0.22	0.33	0.64	0.28	0.39	0.62	0.34	0.44	0.63	0.36	0.46
Process	0.53	0.11	0.19	0.50	0.20	0.28	0.44	0.24	0.31	0.48	0.28	0.35
Task	0.20	0.01	0.01	0.25	0.05	0.08	0.45	0.10	0.17	0.32	0.13	0.19

(a) Results with $K = 128, \lambda = 0.9$, excluding word identity as features.

	Sparse coding only			+Brown			+Orthograpy			+Brown+Orthography		
	P	R	F	P	R	F	P	R	F	P	R	F
Subtask A	0.64	0.25	0.36	0.65	0.27	0.38	0.58	0.33	0.43	0.62	0.34	0.44
Subtask B avg.	0.57	0.22	0.32	0.59	0.25	0.35	0.50	0.29	0.37	0.55	0.30	0.39
Material	0.65	0.26	0.38	0.70	0.31	0.43	0.60	0.35	0.44	0.63	0.36	0.45
Process	0.51	0.21	0.30	0.50	0.22	0.31	0.44	0.25	0.32	0.49	0.29	0.36
Task	0.27	0.05	0.09	0.29	0.04	0.08	0.30	0.14	0.19	0.39	0.11	0.17

(b) Results with $K = 128, \lambda = 0.9$, including word identity as features.

	Sparse coding only			+Brown			+Orthograpy			+Brown+Orthography		
	P	R	F	P	R	F	P	R	F	P	R	F
Subtask A	0.56	0.29	0.38	0.57	0.30	0.40	0.57	0.33	0.42	0.55	0.33	0.41
Subtask B avg.	0.49	0.26	0.34	0.49	0.26	0.34	0.49	0.29	0.36	0.48	0.29	0.36
Material	0.59	0.31	0.40	0.61	0.31	0.41	0.60	0.35	0.44	0.59	0.35	0.44
Process	0.45	0.23	0.30	0.43	0.24	0.30	0.41	0.27	0.33	0.43	0.27	0.33
Task	0.25	0.15	0.19	0.21	0.11	0.14	0.25	0.10	0.14	0.20	0.12	0.15

(c) Results with $K = 1024, \lambda = 0.1$, excluding word identity as features.

	Sparse coding only			+Brown			+Orthograpy			+Brown+Orthography		
	P	R	F	P	R	F	P	R	F	P	R	F
Subtask A	0.56	0.30	0.39	0.59	0.29	0.39	0.58	0.33	0.42	0.58	0.34	0.42
Subtask B avg.	0.49	0.26	0.34	0.52	0.25	0.34	0.50	0.28	0.36	0.50	0.29	0.37
Material	0.65	0.26	0.38	0.70	0.31	0.43	0.60	0.35	0.44	0.63	0.36	0.45
Process	0.44	0.26	0.33	0.50	0.24	0.32	0.42	0.27	0.33	0.44	0.28	0.34
Task	0.21	0.06	0.09	0.18	0.08	0.11	0.24	0.07	0.11	0.20	0.09	0.12

(d) Results with $K = 1024, \lambda = 0.1$, including word identity as features.

Table 4: Ablation experiments on the development set. P=Precision, R=Recall, F=F-scores.

3.2 Results on test data

Based on our experiments on the development data, out official shared task submission employed $K = 128, \lambda = 0.9$ alongside with orthographic and Brown clustering-derived features. One of our official submissions relied on word form features, whereas the other dismissed such ones. The final results of our submissions are included in Table 2.

As our main goal was to verify the applicability of sparse coding derived features in keyphrase extraction as well, we checked the performance of the model which uses all features except for the sparse coding derived ones. The result of that model is presented in Table 3. By comparing these scores with those in Table 2, we can see that even when using a low value for K and a large regularization parameter λ we manage to get better F-scores when sparse coding related features are employed.

4 Conclusion

In this paper, we proposed an approach for extracting keyphrases from scientific publications. A key source of features in our approach were those derived from the sparse coding of continuous word embeddings.

In our approach we did not use any task-specific features (such as lists or gazetters), which implies that i) by relying on some extra task specific features, results could be easily improved on this task and ii) the proposed approach is likely to be successfully applicable to further sequence labeling tasks without severe modifications.

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. Association for Computa-

tional Linguistics, Sofia, Bulgaria, pages 183–192. http://www.aclweb.org/anthology/W13-3520.

Isabelle Augenstein, Mrinal Kanti Das, Sebastian Riedel, Lakshmi Nair Vikraman, and Andrew McCallum. 2017. SemEval 2017 Task 10: ScienceIE - Extracting Keyphrases and Relations from Scientific Publications. In *Proceedings of the International Workshop on Semantic Evaluation*. Association for Computational Linguistics, Vancouver, Canada.

Gábor Berend. 2016a. Exploiting extra-textual and linguistic information in keyphrase extraction. *Natural Language Engineering* 22(1):73–95. https://doi.org/10.1017/S1351324914000126.

Gábor Berend. 2016b. Sparse coding of neural word embeddings for multilingual sequence labeling. *CoRR* abs/1612.07130. http://arxiv.org/abs/1612.07130.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Comput. Linguist.* 18(4):467–479. http://dl.acm.org/citation.cfm?id=176313.176316.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '03, pages 216–223. https://doi.org/10.3115/1119355.1119383.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. SemEval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. ACL, Morristown, NJ, USA, SemEval '10, pages 21–26. http://portal.acm.org/citation.cfm?id=1859664.1859668.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ICML '01, pages 282–289. http://dl.acm.org/citation.cfm?id=645530.655813.

P. Liang. 2005. *Semi-Supervised Learning for Natural Language*. Master's thesis, Massachusetts Institute of Technology.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781. http://arxiv.org/abs/1301.3781.

Naoaki Okazaki. 2007. CRFsuite: a fast implementation of Conditional Random Fields (CRFs). http://www.chokkan.org/software/crfsuite/.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Stroudsburg, PA, USA, CoNLL '09, pages 147–155. http://dl.acm.org/citation.cfm?id=1596374.1596399.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '10, pages 384–394. http://dl.acm.org/citation.cfm?id=1858681.1858721.

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.