PatternRank: Leveraging Pretrained Language Models and Part of Speech for Unsupervised Keyphrase Extraction

Tim Schopf[a], Simon Klimek[b] and Florian Matthes[c]

Department of Informatics, Technical University of Munich, Boltzmannstrasse 3, Garching, Germany

Keywords: Natural Language Processing, Keyphrase Extraction, Pretrained Language Models, Part of Speech.

Abstract: Keyphrase extraction is the process of automatically selecting a small set of most relevant phrases from a given text. Supervised keyphrase extraction approaches need large amounts of labeled training data and perform poorly outside the domain of the training data (Bennani-Smires et al., 2018). In this paper, we present PatternRank, which leverages pretrained language models and part-of-speech for unsupervised keyphrase extraction from single documents. Our experiments show PatternRank achieves higher precision, recall and F_1-scores than previous state-of-the-art approaches. In addition, we present the *KeyphraseVectorizers*[*] package, which allows easy modification of part-of-speech patterns for candidate keyphrase selection, and hence adaptation of our approach to any domain.

1 INTRODUCTION

To quickly get an overview of the content of a text, we can use keyphrases that concisely reflect its semantic context. Keyphrases describe the most essential aspect of a text. Unlike simple keywords, keyphrases do not consist solely of single words, but of several compound words. Therefore, keyphrases provide more information about the content of a text compared to simple keywords. Supervised keyphrase extraction approaches usually achieve higher accuracy than unsupervised ones (Kim et al., 2012; Caragea et al., 2014; Meng et al., 2017). However, supervised approaches require manually labeled training data, which often causes subjectivity issues as well as significant investment of time and money (Papagiannopoulou and Tsoumakas, 2019). In contrast, unsupervised keyphrase extraction approaches do not have these issues and are moreover mostly domain-independent.

Keyphrases and their vector representations are very versatile and can be used in a variety of different Natural Language Processing (NLP) downstream tasks (Braun et al., 2021; Schopf et al., 2022). For example, they can be used as features or input for document clustering and classification (Hulth and Megyesi, 2006; Schopf et al., 2021), they can support extractive summarization (Zhang et al., 2004), or they can be used for query expansion (Song et al., 2006). Keyphrase extraction is particularly relevant for the scholarly domain as it helps to recommend articles, highlight missing citations to authors, identify potential reviewers for submissions, analyze research trends over time, and can be used in many different search scenarios (Augenstein et al., 2017).

In this paper, we present PatternRank, an unsupervised approach for keyphrase extraction based on Pretrained Language Models (PLMs) and Part of Speech (PoS). Since keyphrase extraction is especially important for the scholarly domain, we evaluate PatternRank on a specific dataset from this area. Our approach does not rely on labeled data and therefore can be easily adapted to a variety of different domains. Moreover, PatternRank does not require the input document to be part of a larger corpus, allowing the keyphrase extraction to be applied to individual short texts such as publication abstracts. Figure 1 illustrates the general keyphrase extraction approach of PatternRank.

2 RELATED WORK

Most popular unsupervised keyphrase extraction approaches can be characterized as either statistics-

[a] https://orcid.org/0000-0003-3849-0394

[b] https://orcid.org/0000-0001-8571-7606

[c] https://orcid.org/0000-0002-6667-5452

[*] https://github.com/TimSchopf/KeyphraseVectorizers

Schopf, T., Klimek, S. and Matthes, F.
PatternRank: Leveraging Pretrained Language Models and Part of Speech for Unsupervised Keyphrase Extraction.
DOI: 10.5220/0011546600003335
In *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2022) - Volume 1: KDIR*, pages 243-248
ISBN: 978-989-758-614-9; ISSN: 2184-3228

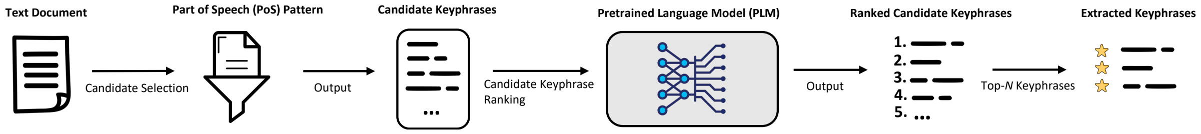

Figure 1: PatternRank approach for unsupervised keyphrase extraction. A single text document is used as input for an initial filtering step where candidate keyphrases are selected which match a defined PoS pattern. Subsequently, the candidate keyphrases are ranked by a PLM based on their semantic similarity to the input text document. Finally, the top-*N* keyphrases are extracted as a concise reflection of the input text document.

based, graph-based, or embedding-based methods, while *Tf-Idf* is a common baseline used for evaluation (Papagiannopoulou and Tsoumakas, 2019).

YAKE uses a set of different statistical metrics including word casing, word position, word frequency, and more to extract keyphrases from text (Campos et al., 2020). *TextRank* uses PoS filters to extract noun phrase candidates that are added to a graph as nodes, while adding an edge between nodes if the words co-occur within a defined window (Mihalcea and Tarau, 2004). Finally, PageRank (Page et al., 1999) is applied to extract keyphrases. *SingleRank* expands the TextRank approach by adding weights to edges based on word co-occurrences (Wan and Xiao, 2008). *RAKE* generates a word co-occurrence graph and assigns scores based on word frequency, word degree, or the ratio of degree and frequency for keyphrase extraction (Rose et al., 2010). Furthermore, Knowledge Graphs can be used to incorporate semantics for keyphrase extraction (Shi et al., 2017). *EmbedRank* leverages Doc2Vec (Le and Mikolov, 2014) and Sent2Vec (Pagliardini et al., 2018) sentence embeddings to rank candidate keyphrases for extraction (Bennani-Smires et al., 2018). More recently, a PLM-based approach was introduced that uses BERT (Devlin et al., 2019) for self-labeling of keyphrases and subsequent use of the generated labels in an LSTM classifier (Sharma and Li, 2019).

3 KEYPHRASE EXTRACTION APPROACH

Figure 1 illustrates the general keyphrase extraction process of our PatternRank approach. The input consists of a single text document which is being word tokenized. The word tokens are then tagged with PoS tags. Tokens whose tags match a previously defined PoS pattern are selected as candidate keyphrases. Then, the candidate keyphrases are fed into a PLM to rank them based on their similarity to the input text document. The PLM embeds the entire text document as well as all candidate keywords as semantic vector representations. Subsequently, the cosine similarities between the document representation and the candidate keyphrase representations are computed and the candidate keyphrases are ranked in descending order based on the computed similarity scores. Finally, the top-*N* ranked keyphrases, which are most representative of the input document, are extracted.

3.1 Candidate Selection with Part of Speech

In previous work, simple noun phrases consisting of zero or more adjectives followed by one or more nouns were used for keyphrase extraction (Mihalcea and Tarau, 2004; Wan and Xiao, 2008; Bennani-Smires et al., 2018). However, we define a more complex PoS pattern to extract candidate keyphrases from the input text document. In our approach, the tags of the word tokens have to match the following PoS pattern in order for the tokens to be considered as candidate keyphrases:

$$\begin{aligned}&\Big(\big(\{.*\}\{HYPH\}\{.*\}\big)\{NOUN\}*\Big)\Big|\\&\Big(\big(\{VBG\}|\{VBN\}\big)?\{ADJ\}*\{NOUN\}+\Big)\end{aligned} \tag{1}$$

The PoS pattern quantifiers correspond to the regular expression syntax. Therefore, we can translate the PoS pattern as *arbitrary parts-of-speech separated by a hyphen, followed by zero or more nouns OR zero or one verb (gerund or present or past participle), followed by zero or more adjectives, followed by one or more nouns*.

3.2 Candidate Ranking with Pretrained Language Models

Earlier work used graphs (Mihalcea and Tarau, 2004; Wan and Xiao, 2008) or paragraph and sentence embeddings (Bennani-Smires et al., 2018) to rank candidate keyphrases. However, we leverage PLMs based on current transformer architectures to rank the candidate keyphrases that have recently demonstrated promising results (Grootendorst, 2020). Therefore, we follow the general EmbedRank (Bennani-Smires et al., 2018) approach for ranking, but use PLMs instead of Doc2Vec (Le and Mikolov, 2014) and Sent2Vec (Pagliardini et al., 2018) to create semantic vector representations of the entire text document as well as all candidate keyphrases. In our experiments, we use SBERT (Reimers and Gurevych, 2019) PLMs since they have been shown to produce state of the art text representations for semantic similarity tasks. Using these semantic vector representations, we rank the candidate keyphrases based on their cosine similarity to the input text document.

4 EXPERIMENTS

In this section, we compare four different approaches for unsupervised keyphrase extraction in the scholarly domain.

4.1 Data

In our experiments, we use the *Inspec* dataset (Hulth, 2003), which consists of 2,000 English computer science abstracts collected from scientific journal articles between 1998 and 2002. Each abstract has assigned two different types of keyphrases. First, controlled and manually assigned keyphrases that appear in the thesaurus of the *Inspec* dataset but do not necessarily have to appear in the abstract. Second, uncontrolled keyphrases that are freely assigned by professional indexers and are not restricted to either the thesaurus or the abstract. In our experiments, we consider the union of both types of keyphrases as the ground truth.

4.2 Evaluation

For evaluation, we compare the performances of four different keyphrase extraction approaches.

YAKE: is a fast and lightweight approach for unsupervised keyphrase extraction from single documents based on statistical features (Campos et al., 2020).

SingleRank: applies a ranking algorithm to word co-occurrence graphs for unsupervised keyphrase extraction from single documents (Wan and Xiao, 2008).

KeyBERT: uses, similar to PatternRank, a PLM to rank candidate keyphrases (Grootendorst, 2020). However, KeyBERT uses simple word n-grams as candidate keyphrases rather than word tokens that match a certain PoS pattern, as in our PatternRank approach. For the KeyBERT experiments, we use the *all-mpnet-base-v2*[1] SBERT model for candidate keyphrase ranking and an n-gram range of $[1,3]$ for candidate keyphrase selection. This means that n-grams consisting of 1, 2 or 3 words are selected as candidate keyphrases.

PatternRank: To select candidate keyphrases, we developed the *KeyphraseVectorizers*[2] package, which allows custom PoS patterns to be defined and returns matching candidate keyphrases. We evaluate two different versions of the PatternRank approach. PatternRank$_{NP}$ selects simple noun phrases as candidate keyphrases and PatternRank$_{PoS}$ selects word tokens whose PoS tags match the pattern defined in section 3.1. In both cases, the *all-mpnet-base-v2* SBERT model is used for candidate keyphrase ranking.

We evaluate the models based on exact match, partial match, and the average of exact and partial match. For each approach, we report Precision@N, Recall@N, and F$_1$@N scores, using the top-N extracted keyphrases respectively. The gold keyphrases always remain the entire set of all manually assigned keyphrases, regardless of N. Additionally, we lowercase the gold keyphrases as well as the extracted keyphrases and remove duplicates. We follow the approach of Rousseau and Vazirgiannis (2015) and calculate Precision@N, Recall@N, and F$_1$@N scores per document and then use the macro-average at the collection level for evaluation. The exact match approach yields true positives only for extracted keyphrases that have an exact string match to one of the gold keyphrases. However, this evaluation approach penalizes keyphrase extraction methods which predict keyphrases that are syntactically different from the gold keyphrases but semantically similar (Rousseau and Vazirgiannis, 2015; Wang et al., 2015). The partial match approach converts gold keyphrases as well as extracted keyphrases to unigrams and yields true positives if the extracted unigram keyphrases have a string match to one of the unigram gold keyphrases (Rousseau and Vazirgiannis, 2015). The drawback of the partial match evaluation approach, however, is that it rewards methods which predict keyphrases that occur in the unigram gold keyphrases but are not appropriate for the corresponding document (Papagiannopoulou and

[1] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[2] https://github.com/TimSchopf/KeyphraseVectorizers

Table 1: Evaluation of our approach against state of the art using the *Inspec* dataset. Precision (P), Recall (R), and F$_1$-score (F$_1$) for N = 5, 10, and 20 are reported. The evaluation results are based on exact match, partial match, and the average of exact and partial match. Two variations of PatternRank are presented: PatternRank$_{NP}$ selects simple noun phrases as candidate keyphrases and PatternRank$_{PoS}$ selects word tokens whose PoS tags match the pattern defined in section 3.1. In both cases, a SBERT PLM is used for candidate keyphrase ranking.

	Method	@5			@10			@20		
		P	R	F$_1$	P	R	F$_1$	P	R	F$_1$
Exact Match	YAKE	26.16	11.71	15.37	20.88	18.45	18.50	16.45	27.78	19.65
	SingleRank	38.11	16.55	21.97	33.29	27.27	28.55	27.24	38.84	30.80
	KeyBERT	12.97	6.08	7.82	11.42	10.53	10.30	9.75	17.14	11.76
	PatternRank$_{NP}$	41.15	18.09	23.92	34.60	28.33	29.66	25.88	36.69	29.19
	PatternRank$_{PoS}$	**41.76**	**18.44**	**24.35**	**36.10**	**29.63**	**30.99**	**27.80**	**39.42**	**31.37**
Partial Match	YAKE	77.45	19.49	29.91	68.20	33.46	42.67	59.69	45.58	48.69
	SingleRank	75.54	19.36	29.56	68.63	33.98	43.24	58.82	53.68	53.68
	KeyBERT	77.48	20.06	30.55	65.78	32.90	41.67	57.11	45.37	48.34
	PatternRank$_{NP}$	**83.64**	**21.93**	**33.29**	**75.27**	**37.62**	**47.69**	62.78	56.69	57.03
	PatternRank$_{PoS}$	82.49	21.61	32.79	74.79	37.50	47.48	**63.21**	**57.66**	**57.71**
Avg. Match	YAKE	51.81	15.60	22.64	44.54	25.96	30.59	38.07	36.68	34.13
	SingleRank	56.83	17.96	25.77	50.96	30.63	35.90	43.03	46.26	42.24
	KeyBERT	45.23	13.07	19.19	38.60	21.72	25.99	33.43	31.23	30.05
	PatternRank$_{NP}$	**62.40**	20.01	**28.61**	54.94	32.98	38.68	44.33	46.69	43.11
	PatternRank$_{PoS}$	62.13	**20.03**	28.57	**55.45**	**33.57**	**39.24**	**45.51**	**48.54**	**44.54**

Tsoumakas, 2019). For empirical comparison of keyphrase extraction approaches, we therefore also report the average of the exact and partial matching results.

The results of our evaluation are shown in Table 1. We can see that our PatternRank approach outperforms all other approaches across all benchmarks. In general, both approaches PatternRank$_{NP}$ and PatternRank$_{PoS}$ perform fairly similarly, whereas PatternRank$_{PoS}$ produces slightly better results in most cases. In the exact match evaluation, PatternRank$_{PoS}$ consistently achieves the best results of all approaches. Furthermore, PatternRank$_{PoS}$ also yields best results in the average mach evaluation for N = 10 and 20. In the partial match evaluation, the PatternRank$_{NP}$ approach marginally outperforms the PatternRank$_{PoS}$ approach and yields best results for N = 5 and 10. However, as we mentioned earlier the partial match evaluation approach, may wrongly reward methods which extract keyphrases that occur in the unigram gold keyphrases but are not appropriate for the corresponding document. Since the PatternRank$_{PoS}$ approach outperforms the PatternRank$_{NP}$ approach in the more important exact match and average match evaluations, we argue that selecting candidate keyphrases based on the PoS pattern defined in Section 3.1 instead of simple noun phrases helps to extract keyphrases predominantly occurring in the scholarly domain. In contrast, skipping the PoS pattern-based candidate keyphrase selection step results in a significant performance decline. KeyBERT uses the same PLM to rank the candidate keyphrases as PatternRank, but uses simple n-grams for candidate keyphrase selection instead of PoS patterns or noun phrases. As a result, the KeyBERT approach consistently performs worst among all approaches. As expected, YAKE was the fastest keyphrase extraction approach because it is a lightweight method based on statistical features. However, the extracted keyphrases are not very accurate and in comparison to PatternRank, YAKE significantly performs worse in all evaluations. SingleRank is the only approach that achieves competitive results compared to PatternRank. Nevertheless, it consistently performs a few percentage points worse than PatternRank across all evaluations. We therefore conclude that our PatternRank achieves state-of-the-art keyphrase extraction results, especially in the scholarly domain.

5 CONCLUSION

We presented the PatternRank approach which leverages PLMs and PoS for unsupervised keyphrase extraction. We evaluated our approach against three different keyphrase extraction methods: one statistics-based approach, one graph-based approach and one PLM-based approach. The results show that the PatternRank approach performs best in terms of precision, recall and F$_1$-score across all evaluations. Furthermore, we evaluated two different PatternRank versions. PatternRank$_{NP}$ selects simple noun phrases as candidate keyphrases and PatternRank$_{PoS}$ selects word tokens whose PoS tags match the pattern defined in Section 3.1. While PatternRank$_{PoS}$ produced better results in the majority of cases, PatternRank$_{NP}$ still performed very well in all benchmarks. We therefore conclude that the PatternRank$_{PoS}$ approach works particularly well in the evaluated scholarly domain. Furthermore, since the use of noun phrases as candidate keyphrases is a more general and domain-independent approach, we propose using PatternRank$_{NP}$ as a simple but effective keyphrase extraction method for arbitrary domains. Future work may investigate how the PLM and PoS pattern used in this approach can be adapted to different domains or languages.

REFERENCES

Augenstein, I., Das, M., Riedel, S., Vikraman, L., and McCallum, A. (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)*, pages 546–555, Vancouver, Canada. Association for Computational Linguistics.

Bennani-Smires, K., Musat, C., Hossmann, A., Baeriswyl, M., and Jaggi, M. (2018). Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 221–229, Brussels, Belgium. Association for Computational Linguistics.

Braun, D., Klymenko, O., Schopf, T., Kaan Akan, Y., and Matthes, F. (2021). The language of engineering: Training a domain-specific word embedding model for engineering. In *2021 3rd International Conference on Management Science and Industrial Engineering*, MSIE 2021, page 8–12, New York, NY, USA. Association for Computing Machinery.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., and Jatowt, A. (2020). Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

Caragea, C., Bulgarov, F. A., Godea, A., and Das Gollapalli, S. (2014). Citation-enhanced keyphrase extraction from research papers: A supervised approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1435–1446, Doha, Qatar. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Grootendorst, M. (2020). Keybert: Minimal keyword extraction with bert.

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, page 216–223, USA. Association for Computational Linguistics.

Hulth, A. and Megyesi, B. B. (2006). A study on automatically extracted keywords in text categorization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 537–544, Sydney, Australia. Association for Computational Linguistics.

Kim, S. N., Medelyan, O., Kan, M.-Y., and Baldwin, T. (2012). Automatic keyphrase extraction from scientific articles.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China. PMLR.

Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., and Chi, Y. (2017). Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 582–592, Vancouver, Canada. Association for Computational Linguistics.

Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking : Bringing order to the web. In *WWW 1999*.

Pagliardini, M., Gupta, P., and Jaggi, M. (2018). Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540, New Orleans, Louisiana. Association for Computational Linguistics.

Papagiannopoulou, E. and Tsoumakas, G. (2019). A review of keyphrase extraction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Rose, S. J., Engel, D. W., Cramer, N., and Cowley, W. (2010). Automatic keyword extraction from individual documents.

Rousseau, F. and Vazirgiannis, M. (2015). Main core retention on graph-of-words for single-document keyword extraction. In Hanbury, A., Kazai, G., Rauber, A., and Fuhr, N., editors, *Advances in Information Retrieval*, pages 382–393, Cham. Springer International Publishing.

Schopf, T., Braun, D., and Matthes, F. (2021). Lbl2vec: An embedding-based approach for unsupervised document retrieval on predefined topics. In *Proceedings of the 17th International Conference on Web Information Systems and Technologies - WEBIST*, pages 124–132. INSTICC, SciTePress.

Schopf, T., Weinberger, P., Kinkeldei, T., and Matthes, F. (2022). Towards bilingual word embedding models for engineering: Evaluating semantic linking capabilities of engineering-specific word embeddings across languages. In *2022 4th International Conference on Management Science and Industrial Engineering (MSIE)*, MSIE 2022, page 407–413, New York, NY, USA. Association for Computing Machinery.

Sharma, P. and Li, Y. (2019). Self-supervised contextual keyword and keyphrase retrieval with self-labelling.

Shi, W., Zheng, W., Yu, J. X., Cheng, H., and Zou, L. (2017). Keyphrase extraction using knowledge graphs. *Data Science and Engineering*, 2:275–288.

Song, M., Song, I. Y., Allen, R. B., and Obradovic, Z. (2006). Keyphrase extraction-based query expansion in digital libraries. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '06, page 202–209, New York, NY, USA. Association for Computing Machinery.

Wan, X. and Xiao, J. (2008). CollabRank: Towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 969–976, Manchester, UK. Coling 2008 Organizing Committee.

Wang, R., Liu, W., and McDonald, C. (2015). Using word embeddings to enhance keyword identification for scientific publications. In Sharaf, M. A., Cheema, M. A., and Qi, J., editors, *Databases Theory and Applications*, pages 257–268, Cham. Springer International Publishing.

Zhang, Y., Zincir-Heywood, N., and Milios, E. (2004). World wide web site summarization.