

Keyphrase Extraction from Scientific Articles via Extractive Summarization

Chrysovalantis-Giorgos Kontoulis and Eirini Papagiannopoulou
and Grigorios Tsoumakas

School of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece
{ckontoul,epapagia,greg}@csd.auth.gr

Abstract

Automatically extracting keyphrases from scholarly documents leads to a valuable concise representation that humans can understand and machines can process for tasks, such as information retrieval, article clustering and article classification. This paper is concerned with the parts of a scientific article that should be given as input to keyphrase extraction methods. Recent deep learning methods take titles and abstracts as input due to the increased computational complexity in processing long sequences, whereas traditional approaches can also work with full-texts. Titles and abstracts are dense in keyphrases, but often miss important aspects of the articles, while full-texts on the other hand are richer in keyphrases but much noisier. To address this trade-off, we propose the use of extractive summarization models on the full-texts of scholarly documents. Our empirical study on 3 article collections using 3 keyphrase extraction methods shows promising results.

1 Introduction

Automatic keyphrase extraction is the process of identifying representative phrases in a document that summarize its content. Keyphrases are important pieces of information for many applications, including information retrieval (Ji et al., 2019; Boudin et al., 2020), text classification (Meng et al., 2019), text summarization (Song et al., 2019), entity recognition (Du et al., 2018) and event detection (Hossny et al., 2020).

This work focuses on keyphrase extraction from scholarly documents. In particular, we consider an interesting issue in this domain, which concerns the part of a scientific article that should be given as input to keyphrase extraction methods.

Table 1 shows representative supervised and unsupervised keyphrase extraction methods from the most popular categories of the task (deep learning,

traditional supervised, graph-based, and statistics-based), along with the parts of academic articles that they consider, among Title+Abstract (TA), Full-text (F) and other Specific Parts (S/P).

| Approaches | TA | F | S/P |
|-----------------------------|----|---|-----|
| Deep Learning | | | |
| Meng et al. (2017) | ✓ | | |
| Basaldella et al. (2018) | ✓ | | |
| Chen et al. (2018) | ✓ | | |
| Ye and Wang (2018) | ✓ | | |
| Wang et al. (2018) | ✓ | | |
| Patel and Caragea (2019) | ✓ | | |
| Chan et al. (2019) | ✓ | | |
| Alzaidy et al. (2019) | ✓ | | |
| Chen et al. (2019) | ✓ | | |
| Çano and Bojar (2019) | ✓ | | |
| Zhu et al. (2020) | ✓ | | |
| Zhou et al. (2020) | ✓ | | |
| Zahedi et al. (2020) | ✓ | | |
| Traditional Supervised | | | |
| Witten et al. (1999) | ✓ | ✓ | |
| Medelyan et al. (2009) | ✓ | ✓ | |
| Nguyen and Luong (2010) | ✓ | ✓ | ✓ |
| Caragea et al. (2014) | ✓ | ✓ | |
| Wang and Li (2017) | ✓ | ✓ | |
| Graph-based | | | |
| Mihalcea and Tarau (2004)* | ✓ | ✓ | |
| Wan and Xiao (2008)* | ✓ | ✓ | |
| Bougouin et al. (2013) | ✓ | ✓ | |
| Sterckx et al. (2015) | ✓ | ✓ | |
| Boudin (2018) | ✓ | ✓ | |
| Mahata et al. (2018) | ✓ | ✓ | |
| Statistics-based | | | |
| <i>TfIdf</i> | ✓ | ✓ | |
| El-Beltagy and Rafea (2009) | ✓ | ✓ | |
| Campos et al. (2020) | ✓ | ✓ | |

Table 1: Types of textual content, i.e., Title+Abstract (TA), Full-text (F), and Specific Parts (S/P) of the document, used by supervised and unsupervised keyphrase extraction approaches in the training and evaluation process. Approaches with an asterisk (*) are evaluated on TAs and Fs in Hasan and Ng (2010).

We can see that recent deep learning keyphrase extraction and generation methods take titles and abstracts as input, due to the complexity in pro-

cessing larger sequences. Traditional supervised learning methods, as well as unsupervised ones can handle full-texts, but this does not necessarily lead to better results compared to using just titles and abstracts. Papagiannopoulou and Tsoumakas (2018) show that graph-based methods achieve better accuracy when titles and abstracts are used, while the strong baseline TfIdf works best with full-text. Florescu and Caragea (2017) and Boudin (2018) show that keyphrases generally occur in positions very close to the beginning of a scholarly document. Nguyen and Luong (2010) show that title and abstracts have the highest density of keyphrases, followed by the conclusions, introduction and related work sections.

It appears that there is a trade-off between using titles and abstracts versus using full-texts of academic papers as input to keyphrase extraction methods. Full-texts provide richer information, including more keyphrases, but at the same time they are much more noisy compared to the titles and abstracts. Motivated from this observation, our scientific question is whether using automated summarization models on the full-text of a scientific article can lead to textual information that is richer than titles and abstracts, yet less noisy than full-texts.

Towards answering this question, we present some first steps employing extractive summarization. Our main goals are to: a) investigate the dynamics of summarization in keyphrase extraction, paving the way for the research community to develop approaches combining techniques from both tasks (e.g., via multi-task learning) and b) provide some guidelines to practitioners of the field suggesting better utilization of the full-texts. Our empirical study provides strong evidence that the full-text extractive summaries manage to capture keyphrases, which in most cases improve the performance of state-of-the-art supervised and unsupervised keyphrase extraction methods (regarding the F_1 score) on three datasets compared to the conventional use of abstracts and full-texts.

2 Our Approach and Alternatives

We are interested in finding out whether we can improve the signal-to-noise ratio of the input given to keyphrase extraction approaches by applying automated summarization on the full-text of scientific articles. As a first step towards investigating this hypothesis, we focus on *extractive summarization*

models.

We generate extractive summaries from the corresponding full-texts using the pre-trained distilled RoBERTa model *distilroberta-base-ext-sum* from the TransformerSum¹ library. Distilled RoBERTa is a version of *RoBERTa* (Liu et al., 2019), which is based on DistilBERT (Sanh et al., 2019). It is a lighter, faster and smaller variant of the original RoBERTa, that achieves a time speed-up of 50%, while retaining 95% performance of the original model.

Furthermore, we investigate the utility of alternative input types, such as the first three paragraphs of the document that include the title, the abstract and a part of the document’s introduction. We experiment with two different paragraph lengths in words, i.e., 220 and 400.

Our investigation includes the standard input types, i.e., title+abstract and full-text, too. For deep learning methods, we split full-texts into sentences and paragraphs, as they cannot handle their whole length at once due to memory limitations.

Finally, we explore an ensemble approach to keyphrase extraction, which involves the late fusion of two input types: the standard title plus abstract and the title plus the extractive summary. We apply keyphrase extraction methods to these two input types independently and then consider the union of the extracted keyphrases.

Table 2 presents all these approaches along with their abbreviations, which will be used in the rest of our work.

| Abbr. | Description |
|-------------------|---|
| TA | Abstract |
| ABSE | Abstract in Sentences |
| F | Full-text |
| FP | Full-text in Paragraphs |
| FS | Full-text in Sentences |
| TS | Extractive Summary |
| AS | Abstract \cup Extractive Summary |
| 3P ₂₂₀ | First 3 Paragraphs - length in words: 220 |
| 3P ₄₀₀ | First 3 Paragraphs - length in words: 400 |

Table 2: Descriptions of the different approaches along with their abbreviations. The title is part of the input in all cases.

3 Experimental Setup

Our empirical study includes three keyphrase extraction methods: TfIdf, as a baseline method, MultipartiteRank (MR) (Boudin, 2018), as a strong

¹<https://github.com/HHousen/TransformerSum>

graph-based method, and Bi-LSTM-CRF (BLC) (Alzaidy et al., 2019), as a strong neural model. Due to the lack of publicly available code for a BLC model tailored to keyphrase extraction, we proceeded to our own implementation, which we make publicly available along with all experiments in this paper².

BLC is trained using the train and validation sets from (Meng et al., 2017). Specifically, we trained both models described in (Alzaidy et al., 2019), i.e., the BLC_{TA} on the documents’ abstracts and the BLC_{ABSE} on the abstracts’ sentences (used only with test datasets that their text is split in sentences and only for the model comparison). Experiments were performed on a Ryzen 5 3600 CPU with 16GB RAM. Training the model on title and abstract takes approximately 24 hours for a total of 5 epochs, while training on title and abstract split in sentences takes about 5 hours to complete.

These keyphrase extraction methods are evaluated on three well-known datasets that contain full-text articles from the computer science domain: SemEval (Kim et al., 2010), NUS (Nguyen and Kan, 2007), and ACM (Krapivin et al., 2008). These datasets contain 244, 211, and 2304 documents, respectively (we merged the train and test sets of the SemEval dataset).

We compute F_1 ($F_1@10$ for unsupervised methods) according to both the exact (E) and partial (P) (Rousseau and Vazirgiannis, 2015) string match to determine the number of correctly matched phrases with the golden ones for a document. We also apply stemming to the methods’ output and the article’s golden phrases as a pre-processing step before the evaluation process. We employ the authors’ and readers’ (in case they are available) keyphrases as a gold evaluation standard for all dataset collections.

Finally, we use a two-sided Wilcoxon signed-rank test to check the statistical significance of the results in terms of the most popular exact match evaluation between the proposed input types and the conventional ones, at a significance level of 0.05. We denote with a “*” the statistical significance with TA and with a “†” the statistical significance with ABSE or F (in cases there is an improvement).

²<https://github.com/intelligence-csd-auth-gr/keyphrase-extraction-via-summarization>

4 Results and Discussion

Table 3 gives the percentage and actual number (in parentheses) of keyphrases that appear inside each textual content type (F, $3P_{400}$, $3P_{220}$, TS, TA) for each of the 3 datasets (SemEval, NUS, ACM). We can see that full-texts contain the highest percentage of keyphrases, as expected. Note that this number is less than 1, as a small percentage of the keyphrases that authors or readers assign to papers do not appear inside the paper’s full-text. The percentages of $3P_{400}$ and $3P_{220}$ are high too. Extractive summaries contain less keyphrases than the previous content types, but more than titles+abstracts. This is a positive sign, which combined with low amount of noise, could lead to improved keyphrase extraction results.

| | SemEval | NUS | ACM |
|----------------|--------------|--------------|--------------|
| F | 0.857 (3239) | 0.878 (2157) | 0.738 (9079) |
| $3P_{400}$ | 0.668 (2523) | 0.696 (1710) | 0.665 (8172) |
| $3P_{220}$ | 0.582 (2197) | 0.624 (1533) | 0.616 (7572) |
| TS | 0.518 (1956) | 0.576 (1415) | 0.573 (7041) |
| TA | 0.439 (1658) | 0.514 (1264) | 0.530 (6518) |
| Total $_{KPs}$ | 3778 | 2458 | 12296 |

Table 3: Percentage of keyphrases, along with actual number of keyphrases inside parentheses, that are found in each textual content type (TA, F, TS, $3P_{220}$, $3P_{400}$) for each of the 3 datasets (SemEval, NUS, ACM). The last row shows the total number of keyphrases per dataset (Total $_{KPs}$).

One disadvantage of extractive summaries, is that they require an additional pre-processing step compared to the rest pre-existing textual content types. The average time to generate the extractive summary per document in the machine used for the experiments is 2.21, 2.13, and 2.34 seconds for the SemEval, NUS, and ACM datasets, respectively. This is not high for offline applications, while for online ones, higher scale hardware and/or more efficient architectures could be employed.

4.1 Bi-LSTM-CRF

Table 4 shows the results of our implementation of the BLC model, along with the ones published in (Alzaidy et al., 2019) for the kp20k test set from (Meng et al., 2017). BLC solves a sequence classification task: for each word, it outputs a binary label indicating whether this word belongs to a keyphrase or not. The evaluation of BLC in (Alzaidy et al., 2019) was based on the F_1 -score of this binary sequence classification task that BLC

solves, which we also compute for our implementation. We also show the results of our implementation in terms of the exact and partial evaluation approaches.

| | S | E | P |
|-----------------------|-------|-------|-------|
| Our BLC_{TA} | 0.381 | 0.137 | 0.408 |
| Original BLC_{TA} | 0.418 | - | - |
| Our BLC_{ABSE} | 0.288 | 0.150 | 0.301 |
| Original BLC_{ABSE} | 0.356 | - | - |

Table 4: F_1 based on sequence (S), exact (E) and partial (P) evaluation for the original BLC approach and our implementation.

The results of the two BLC_{TA} implementations are close to each other. The difference could be attributed to two things: a) the pre-processing of the data, which is not described in detail in (Alzaidy et al., 2019), and b) the fact that Alzaidy et al. (2019) might have not included the title in their experiments, as this is not clear in the paper. For BLC_{ABSE} , the difference is larger which might be a result of the above and the selected hyperparameters, which we fine-tuned on BLC_{TA} .

Table 5 shows the results of BLC with the standard and proposed input types. Results indicate no significant improvement using extractive summaries compared to titles and abstracts, even though TS includes more keyphrases across all datasets (see Table 3). However, this evaluation may be slightly unfair to TS as input to BLC, since the model used the original documents’ abstracts for training. TAs and TSs may have substantial differences in their syntax, structure, etc. Nevertheless, AS performs better than TA, meaning that TS manages to introduce unseen keyphrases to TA, which seems promising for the potential of extractive summarization.

| BLC | SemEval | | NUS | | ACM | |
|------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | E | P | E | P | E | P |
| TA | 0.103 | 0.196 | 0.129 | 0.270 | 0.148 | 0.325 |
| ABSE | 0.161 | 0.325 | 0.182 | 0.360 | 0.179 | 0.387 |
| FP | 0.157* | 0.349 | 0.144* | 0.319 | 0.082 | 0.241 |
| FS | 0.132* | 0.316 | 0.102 | 0.226 | 0.068 | 0.175 |
| TS | 0.097 | 0.192 | 0.128 | 0.265 | 0.139 | 0.317 |
| AS | 0.118* | 0.226 | 0.145 | 0.300 | 0.151* | 0.345 |
| $3P_{220}$ | 0.143* | 0.264 | 0.168* | 0.337 | 0.157* | 0.352 |
| $3P_{400}$ | 0.088 | 0.187 | 0.102 | 0.239 | 0.138 | 0.336 |

Table 5: F_1 based on exact (E) and partial (P) evaluation approach for BLC on 3 different datasets (SemEval, NUS, ACM) using various textual content types as input, i.e., TA, ABSE, FP, FS, TS, AS, $3P_{220}$, $3P_{400}$.

In addition, our findings show that we achieve higher F_1 -scores when we predict on the abstracts split into sentences rather than the entire abstract. This indicates the inability of the model to retain past information from longer text excerpts, which is a common problem for RNNs. Note that for all the results of the experiments in Table 5, we utilize only the BLC_{TA} model, even on the text excerpts split in sentences as it showed superior performance than the BLC_{ABSE} .

Moreover, FP and $3P_{220}$ seem to be better alternatives to TA, as they constitute richer sources in keyphrases, and the trained BLC_{TA} model can utilize them properly. Finally, the FS approach fails to detect the full-text’s keyphrases due to the combination of noise and the disparity of important context, which is a result of the extreme fragmentation of long texts to sentences.

4.2 Unsupervised methods

Tables 6 and 7 show that the unsupervised methods TfIdf and MR certainly benefit from the extractive summaries (TS) as they outperform the conventional approaches (TA, F) (except for the MR method on NUS where the TS’s F_1 -score is slightly lower than the F’s one). $3P_{200}$ and $3P_{400}$ approaches, in most cases, do not improve the corresponding methods’ accuracy. Although the introductory parts of a document contain many keyphrases, they are also quite noisy due to general descriptions related to the document’s topics.

| TfIdf | SemEval | | NUS | | ACM | |
|------------|----------------------------|--------------|---------------|--------------|----------------------------|--------------|
| | E | P | E | P | E | P |
| TA | 0.143 | 0.312 | 0.179 | 0.377 | 0.129 | 0.351 |
| F | 0.140 | 0.289 | 0.193 | 0.347 | 0.112 | 0.285 |
| TS | 0.162* [†] | 0.325 | 0.201* | 0.388 | 0.143* [†] | 0.361 |
| AS | 0.160* [†] | 0.349 | 0.190 | 0.393 | 0.129 | 0.349 |
| $3P_{220}$ | 0.134 | 0.325 | 0.139 | 0.317 | 0.083 | 0.245 |
| $3P_{400}$ | 0.160* [†] | 0.362 | 0.171 | 0.361 | 0.099 | 0.277 |

Table 6: $F_1 @ 10$ based on exact (E) and partial (P) evaluation approach for TfIdf on 3 different datasets (SemEval, NUS, ACM) using various textual content types as input, i.e., TA, F, TS, AS, $3P_{220}$, $3P_{400}$.

5 Conclusions and Future Work

Our work set out to investigate whether using automated summarization, as a pre-processing step, can lead to improved results in the task of keyphrase extraction from scholarly documents. Our empirical study shows that unsupervised approaches improve

| MR | SemEval | | NUS | | ACM | |
|-------------------|---------------------------|--------------|--------------|--------------|--------------------------|--------------|
| | E | P | E | P | E | P |
| TA | 0.137 | 0.344 | 0.154 | 0.376 | 0.116 | 0.354 |
| F | 0.135 | 0.343 | 0.158 | 0.396 | 0.100 | 0.333 |
| TS | 0.145 | 0.358 | 0.157 | 0.383 | 0.117[†] | 0.360 |
| AS | 0.150^{*†} | 0.367 | 0.158 | 0.376 | 0.110 [†] | 0.339 |
| 3P ₂₂₀ | 0.128 | 0.335 | 0.125 | 0.309 | 0.077 | 0.247 |
| 3P ₄₀₀ | 0.134 | 0.351 | 0.135 | 0.324 | 0.083 | 0.261 |

Table 7: $F_1 @ 10$ based on exact (E) and partial (P) evaluation approach for MR on 3 different datasets (SemEval, NUS, ACM) using various input types, i.e., TA, F, TS, AS, 3P₂₂₀, 3P₄₀₀.

their accuracy using extractive summaries as input, highlighting the full-text’s useful information for the task and showing a positive relationship between the tasks of extractive summarization and keyphrase extraction.

It is worth noting that even though the gains on the exact match F_1 -scores seem to be moderate, this does not necessarily reflect the actual performance gain. Considering that exact match scores are generally low due to the strict nature of the method, a moderate increase in performance leads to considerable percentage gain over the initial performance.

As future work, an interesting direction would be to experiment with additional summarization methods, including abstractive ones as well as their combination with extractive ones. In addition, we could experiment with additional recent and state-of-the-art keyphrase extraction methods, including methods building on top of contextual embeddings (Sahrawat et al., 2020).

References

Rabah Alzaidy, Cornelia Caragea, and C. Lee Giles. 2019. [Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents](#). In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 2551–2557. ACM.

Marco Basaldella, Elisa Antolli, Giuseppe Serra, and Carlo Tasso. 2018. [Bidirectional LSTM recurrent neural network for keyphrase extraction](#). In *Digital Libraries and Multimedia Archives - 14th Italian Research Conference on Digital Libraries, IRCDL 2018, Udine, Italy, January 25-26, 2018, Proceedings*, volume 806 of *Communications in Computer and Information Science*, pages 180–187. Springer.

Florian Boudin. 2018. [Unsupervised keyphrase extraction with multipartite graphs](#). In *Proceedings of the 2018 Conference of the North American Chapter of*

the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 667–672, New Orleans, Louisiana. Association for Computational Linguistics.

Florian Boudin, Ygor Gallina, and Akiko Aizawa. 2020. [Keyphrase generation for scientific document retrieval](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1118–1126, Online. Association for Computational Linguistics.

Adrien Bouguin, Florian Boudin, and Béatrice Daille. 2013. [TopicRank: Graph-based topic ranking for keyphrase extraction](#). In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 543–551, Nagoya, Japan. Asian Federation of Natural Language Processing.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. [Yake! keyword extraction from single documents using multiple local features](#). *Inf. Sci.*, 509:257–289.

Erion Çano and Ondřej Bojar. 2019. [Keyphrase generation: A text summarization struggle](#). 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 666–672, Minneapolis, Minnesota. Association for Computational Linguistics.

Cornelia Caragea, Florin Adrian Bulgarov, Andreea Godea, and Sujatha Das Gollapalli. 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.

Hou Pong Chan, Wang Chen, Lu Wang, and Irwin King. 2019. [Neural keyphrase generation via reinforcement learning with adaptive rewards](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2163–2174, Florence, Italy. Association for Computational Linguistics.

Jun Chen, Xiaoming Zhang, Yu Wu, Zhao Yan, and Zhoujun Li. 2018. [Keyphrase generation with correlation constraints](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4057–4066, Brussels, Belgium. Association for Computational Linguistics.

Wang Chen, Yifan Gao, Jiani Zhang, Irwin King, and Michael R. Lyu. 2019. [Title-guided encoding for keyphrase generation](#). In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA,*

- January 27 - February 1, 2019, pages 6268–6275. AAAI Press.
- Jingcheng Du, Yaoyun Zhang, Jianhong Luo, Yuxi Jia, Qiang Wei, Cui Tao, and Hua Xu. 2018. [Extracting psychiatric stressors for suicide from social media using deep learning](#). *BMC Medical Informatics Decis. Mak.*, 18(S-2):77–87.
- Samhaa R. El-Beltagy and Ahmed A. Rafea. 2009. [Kp-miner: A keyphrase extraction system for english and arabic documents](#). *Inf. Syst.*, 34(1):132–144.
- Corina Florescu and Cornelia Caragea. 2017. [Position-Rank: An unsupervised approach to keyphrase extraction from scholarly documents](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1115, Vancouver, Canada. Association for Computational Linguistics.
- Kazi Saidul Hasan and Vincent Ng. 2010. [Conundrums in unsupervised keyphrase extraction: Making sense of the state-of-the-art](#). In *COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China*, pages 365–373. Chinese Information Processing Society of China.
- Ahmad Hany Hossny, Lewis Mitchell, Nick Lothian, and Grant Osborne. 2020. [Feature selection methods for event detection in twitter: a text mining approach](#). *Soc. Netw. Anal. Min.*, 10(1):61.
- Xiaonan Ji, Han-Wei Shen, Alan Ritter, Raghu Machiraju, and Po-Yin Yen. 2019. [Visual exploration of neural document embedding in information retrieval: Semantics and feature selection](#). *IEEE Trans. Vis. Comput. Graph.*, 25(6):2181–2192.
- 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*, pages 21–26, Uppsala, Sweden. Association for Computational Linguistics.
- Mikalai Krapivin, Aliaksandr Autayeu, and Maurizio Marchese. 2008. Large dataset for keyphrases extraction. In *Technical Report DISI-09-055*. Trento, Italy.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692.
- Debanjan Mahata, John Kuriakose, Rajiv Ratn Shah, and Roger Zimmermann. 2018. [Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 634–639. Association for Computational Linguistics.
- Olena Medelyan, Eibe Frank, and Ian H. Witten. 2009. [Human-competitive tagging using automatic keyphrase extraction](#). In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1318–1327, Singapore. Association for Computational Linguistics.
- Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 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.
- Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2019. [Weakly-supervised hierarchical text classification](#). In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6826–6833. AAAI Press.
- Rada Mihalcea and Paul Tarau. 2004. [TextRank: Bringing order into text](#). In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, pages 404–411. ACL.
- Thuy Dung Nguyen and Min-Yen Kan. 2007. [Keyphrase extraction in scientific publications](#). In *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers, 10th International Conference on Asian Digital Libraries, ICADL 2007, Hanoi, Vietnam, December 10-13, 2007, Proceedings*, volume 4822 of *Lecture Notes in Computer Science*, pages 317–326. Springer.
- Thuy Dung Nguyen and Minh-Thang Luong. 2010. [WINGNUS: Keyphrase extraction utilizing document logical structure](#). In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 166–169, Uppsala, Sweden. Association for Computational Linguistics.
- Eirini Papagiannopoulou and Grigorios Tsoumakas. 2018. [Local word vectors guiding keyphrase extraction](#). *Inf. Process. Manag.*, 54(6):888–902.
- Krutarth Patel and Cornelia Caragea. 2019. [Exploring word embeddings in crf-based keyphrase extraction from research papers](#). In *Proceedings of the 10th International Conference on Knowledge Capture, K-CAP 2019, Marina Del Rey, CA, USA, November 19-21, 2019*, pages 37–44. ACM.
- François Rousseau and Michalis Vazirgiannis. 2015. [Main core retention on graph-of-words for single-document keyword extraction](#). In *Advances in Information Retrieval - 37th European Conference on IR*

- Research, ECIR 2015, Vienna, Austria, March 29 - April 2, 2015. Proceedings*, volume 9022 of *Lecture Notes in Computer Science*, pages 382–393.
- Dhruva Sahrawat, Debanjan Mahata, Haimin Zhang, Mayank Kulkarni, Agniv Sharma, Rakesh Gosangi, Amanda Stent, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2020. [Keyphrase extraction as sequence labeling using contextualized embeddings](#). In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part II*, volume 12036 of *Lecture Notes in Computer Science*, pages 328–335. Springer.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. [Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter](#). *CoRR*, abs/1910.01108.
- Shengli Song, Haitao Huang, and Tongxiao Ruan. 2019. [Abstractive text summarization using LSTM-CNN based deep learning](#). *Multim. Tools Appl.*, 78(1):857–875.
- Lucas Sterckx, Thomas Demeester, Johannes Deleu, and Chris Develder. 2015. [Topical word importance for fast keyphrase extraction](#). In *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*, pages 121–122. ACM.
- Xiaojun Wan and Jianguo Xiao. 2008. [Single document keyphrase extraction using neighborhood knowledge](#). In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 855–860. AAAI Press.
- Liang Wang and Sujian Li. 2017. [PKU_ICL at SemEval-2017 task 10: Keyphrase extraction with model ensemble and external knowledge](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 934–937, Vancouver, Canada. Association for Computational Linguistics.
- Yanan Wang, Qi Liu, Chuan Qin, Tong Xu, Yijun Wang, Enhong Chen, and Hui Xiong. 2018. [Exploiting topic-based adversarial neural network for cross-domain keyphrase extraction](#). In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*, pages 597–606. IEEE Computer Society.
- Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. [KEA: practical automatic keyphrase extraction](#). In *Proceedings of the Fourth ACM conference on Digital Libraries, August 11-14, 1999, Berkeley, CA, USA*, pages 254–255. ACM.
- Hai Ye and Lu Wang. 2018. [Semi-supervised learning for neural keyphrase generation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4142–4153, Brussels, Belgium. Association for Computational Linguistics.
- Amin Ghazi Zahedi, Morteza Zahedi, and Mansoor Fateh. 2020. [A deep extraction model for an unseen keyphrase detection](#). *Soft Comput.*, 24(11):8233–8242.
- Tao Zhou, Yuxiang Zhang, and Haoxiang Zhu. 2020. [Multi-level memory network with crfs for keyphrase extraction](#). In *Advances in Knowledge Discovery and Data Mining - 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11-14, 2020, Proceedings, Part I*, volume 12084 of *Lecture Notes in Computer Science*, pages 726–738. Springer.
- Xun Zhu, Chen Lyu, Donghong Ji, Han Liao, and Fei Li. 2020. [Deep neural model with self-training for scientific keyphrase extraction](#). *Plos one*, 15(5):e0232547.