PubMed Author-assigned Keyword Extraction (PubMedAKE) Benchmark

Jiasheng Sheng
shengjiasheng2000@gmail.com
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

Zelalem Gero
Microsoft Research
Redmond, Washington, USA
zelalemgero@microsoft.com

Joyce C. Ho
Emory University
Atlanta, USA
joyce.c.ho@emory.edu

ABSTRACT

With the ever-increasing abundance of biomedical articles, improving the accuracy of keyword search results becomes crucial for ensuring reproducible research. However, keyword extraction for biomedical articles is hard due to the existence of obscure keywords and the lack of a comprehensive benchmark. PubMedAKE is an author-assigned keyword extraction dataset that contains the title, abstract, and keywords of over 843,269 articles from the PubMed open access subset database. This dataset, publicly available on Zenodo, is the largest keyword extraction benchmark with sufficient samples to train neural networks. Experimental results using state-of-the-art baseline methods illustrate the need for developing automatic keyword extraction methods for biomedical literature.

CCS CONCEPTS

• **Applied computing** → *Bioinformatics*; **Health informatics**.

KEYWORDS

datasets, PubMed literature, keyphrases extraction, keywords extraction

ACM Reference Format:
Jiasheng Sheng, Zelalem Gero, and Joyce C. Ho. 2022. PubMed Author-assigned Keyword Extraction (PubMedAKE) Benchmark. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557675

1 INTRODUCTION

The rapid growth of biomedical literature makes searching for specific articles difficult. As a motivating example, PubMed Central (PMC) is a popular digital repository for biomedical and life science journals and contains more than 7.5 million articles [12]. PMC is often used to retrieve articles for systematic reviews and is a crucial component for evidence-based medicine [42]. While PMC uses Medical Subject Headings (MeSH), a controlled vocabulary thesaurus, to index articles and make finding similar documents easier, there are two major limitations: (1) users must be familiar with the subject headings and (2) the terms may not fully reflect the authors' intentions. An alternative to the MeSH terms is to use author-assigned keywords to summarize the articles. Although the majority of the MeSH terms' meanings are covered or closely related to author keywords [6], the majority of the MeSH terms do not match the author keywords. Using the *PubMedAKE*, we evaluated the partial match between MeSH terms and author keywords, and the MeSH terms only achieved an F1 score of 0.048.

Author-assigned keywords are often used as a proxy for expert annotations and serve as the reference evaluation for many automatic keyphrase extraction benchmark datasets including emails, computer science articles, and news articles [17, 22, 27, 28]. Despite the lack of consistency and standardization across articles, the author-assigned keywords are often correlated with the standardized descriptors assigned by professional indexers [21]. As such, considerable research in automatic keyphrase extraction has been done in the general domain towards summarizing articles using author-assigned keywords to express the crucial aspects of the content [4, 42]. There are various datasets for evaluating automatic keyphrase extraction that encompasses scientific articles, emails, news articles, and social media including a large curated set of 17 benchmark datasets[1]. Although there are several abstract-based datasets in the benchmark, only KP20k [17] has sufficient samples to train a neural network. While state-of-the-art keyphrase extraction models achieve reasonable performance on paper abstracts, scientific articles, and news articles, their performance generally suffers when applied to biomedical literature [18, 19, 20].

The task of identifying author keywords in biomedical literature has been done previously [18, 20], yet they rely on three small-scale datasets. Table 1 summarizes the existing abstract-based keyphrase datasets and the PubMed-based keyphrase datasets. Moreover, existing works predominantly focus solely on extractive keyphrase detection, or identification of words present in the title or abstract, and ignore abstractive keyphrase extraction, or identification of words not present in the title or abstract. Finally, the lack of a standardized biomedical article dataset is problematic as there is an abundance of subject-specific terminologies that prevents existing state-of-the-art keyphrase extraction algorithms to generalize to this domain. Therefore, the goal of this work is to create a new standardized extractive and abstractive dataset, *PubMedAKE*, for evaluating author keyphrase extraction on PubMed articles.

Our approach is to construct an author-assigned keyword dataset using the entire PubMed Open Access Subset. *PubMedAKE* has two key differences compared to the existing PubMed keyphrase extraction datasets. First, we identify both the extractive and abstractive author-assigned keyphrases. Second, we do not restrict the dataset

CIKM '22, October 17–21, 2022, Atlanta, GA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557675

[1] https://github.com/boudinfl/ake-datasets

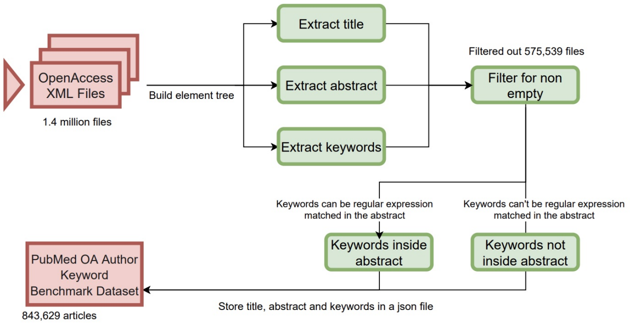
OpenAccess XML Files
1.4 million files
Build element tree
Extract title
Extract abstract
Extract keywords
Filtered out 575,539 files
Filter for non empty
Keywords can be regular expression matched in the abstract
Keywords can't be regular expression matched in the abstract
Keywords inside abstract
Keywords not inside abstract
PubMed OA Author Keyword Benchmark Dataset
843,629 articles
Store title, abstract and keywords in a json file

Figure 1: A flow chart of the data extraction methodology.

Dataset	Train	Test	# words	# kp	% abs
KP20k [28]	527090	20000	176.0	5.3	42.6
PubMed*† [38]	-	1320	5323.0	5.4	16.9
NamedKeys* [20]	-	3049	206.5	14.3	0.0
WWW [13]	-	1330	163.5	4.8	52.0
Inspec [22]	1000	500	134.6	9.8	22.4
Biomedical*¶ [7]	-	1799	-	5.31	57.8
PubMedAKE	505959	168653	216.9	5.2	41.6

Table 1: Statistics of existing scholarly keyphrase datasets and our proposed dataset. †, ∗, and ¶ denote full-text (instead of abstract), articles from PubMed, and not publicly released. The table summarizes the average number of keyphrases (# kp) and words (# words) per document and the ratio of abstractive keyphrases (% abs).

to contain only articles with a specific topic or number of keywords. Figure 1 illustrates the process used to create *PubMedAKE*, which contains 843,629 article abstracts. *PubMedAKE* does not contain the contents of the entire article, which can provide a more holistic view of the article [37], but may result in worse keyphrase extraction performance [31]. It is important to highlight that *PubMedAKE* serves as the **largest keyphrase extraction dataset** to date, a 49% increase over KP20k [28].

2 DATASET CONSTRUCTION

PubMedAKE is constructed from all the non-commercial use articles in the PubMed Open Access Subset, which consisted of 1.4 million files. Each XML file is individually parsed using a customized version of the PubMed parser [1] to extract the title, abstract, and keyphrases. An article is excluded from the dataset if there is no title, abstract, or author-assigned keyphrases. Figure 1 provides an overview of the dataset construction process.

2.1 Title and Abstract Extraction

The title is obtained using the `<article-title>` XML tag. The text is then standardized by ignoring any special formatting tokens by replacing any tabs with the space character and removal of any bold and italic symbols. The abstract is parsed in the same manner by first extracting all paragraphs inside the `<abstract>` XML tag and then ignoring the same special formatting characters. Tabs and new line characters are normalized using the space representation, while bold and italic symbols are stripped from the text. The entire title and abstract are then normalized using ASCII lowercase encoding.

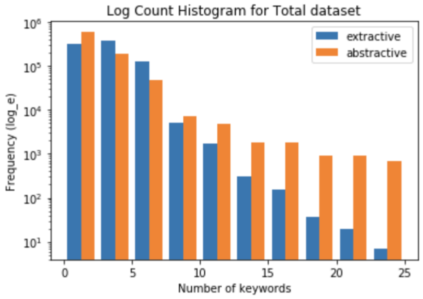

Figure 2: A histogram of the number of keywords in *PubMedAKE*. The x axis ranges between 1 to 25 keywords as there are a limited number of articles with more than 25 keywords, with a maximum of 564.

2.2 Keyword Extraction

The list of keywords are extracted by identifying all the keyword groups (i.e., `<kwd-group>` or`<keyword-group>` tags). The keywords are then identified within these groups using the `<kwd>` tag. It is important to note that there might be multiple keyword groups in one XML document. Thus, our process finds all such groups and extracts the keywords correspondingly. Any special formatting tokens including the newline character, the tab character, any italic labels, or any bold labels are stripped from the keyword. The keyword list is then filtered into two sets: (1) extractive keyphrases, or those that appear inside the abstract, and (2) abstractive keyphrases, or those that do not appear inside the abstract.

2.3 Data Release

PubMedAKE contains 843,629 articles that have at least one keyword, a title, and an abstract. The dataset is stored in the Javascript Object Notation (JSON) format, a common format for existing keyword benchmarks [17, 20]. Each document is indexed using the PubMed id (e.g., "PMC24102982") which serves as a seamless reference to the original XML file (e.g., PMC24102982.xml). Each object then contains the following key / value pairs: (1) the title ("title"), (2) the abstract contents ("abstract"), (3) the extractive keywords ("keywords_in") and (4) the abstractive keywords ("keywords_not_in").

Figure 2 shows the distribution of both the extractive and abstractive keywords in *PubMedAKE*. It is important to note that unlike existing studies [20], there is no threshold range for the number of keywords. Thus, the minimum and the maximum number are 1 and 564. The average number of keyphrases per article is 3 and 2 for extractive and abstractive, respectively. Each article contains approximately 14 words in the title and 217 words in the abstract.

A random partition of 6:2:2 was used to obtain the train, validation, and test sets of *PubMedAKE*. `train.json`, `validate.json`, `test.json` contains 505959, 168653, and 168634 documents, respectively. This dataset is the largest benchmark and 49% larger than the KP20k dataset [28]. Given the distribution of keywords, we also created a selected set, *PubMedAKE*$_{small}$ that contains between 5 and 25 extractive keywords. The smaller dataset, `small_train.json`, `small_validate.json`, `small_test.json` contains 82011, 27336, and 27336 articles respectively. The entire curated dataset is available on Zenodo[2].

The code that is used to create *PubMedAKE*, demonstrate the abstractive and extractive keyphrase extraction tasks, and evaluate the algorithms is available on GitHub under the BSD-2-Clause license[3]. The GitHub repository also details the data structures and examples of the extracted keywords. A small sample of 1000 articles is directly available in the GitHub repository to facilitate algorithmic development.

3 BENCHMARK EVALUATION

Consistent and strong baseline models are necessary to compare new keyphrase extraction algorithms. There are a variety of state-of-the-model algorithms that exist in the general domain keyphrase extraction algorithms. Yet, not all the same baseline models are used, nor do experiments differentiate between extractive and abstractive tasks. As an example, the large-scale evaluation of keyphrase extraction models on nine benchmark datasets combined both together [17]. Here, we separate model assessment into extractive keyphrase and abstractive keyphrase evaluation.

Existing works can be categorized as either unsupervised or supervised approaches. Within the supervised approaches, models are typically classified into whether or not they rely on a neural network. The neural network based models have yielded better performance and include using an encoder-decoder architecture [14, 15, 28, 41] or a long short term memory network with a conditional random field [3, 18, 35, 36]. Within the unsupervised approaches, the popular approaches are either graph-based [9, 10, 16, 30, 40, 43] or statistical-based [5, 11, 34].

3.1 Evaluation Metrics

While the best metric for assessing keyphrase extraction performance is debatable, a common strategy is to compare the top k extracted keyphrases against the ground truth keywords. Thus, we assess models based on precision, recall, and F1 on the top 5, 10, and 15 extracted keyphrases. Precision captures the ratio between the correctly identified keywords out of the total number of extracted keyphrases while recall captures the ratio between the correctly identified keyphrases out of the true author-assigned keywords. The f-measure is then the harmonic mean of recall and precision.

A common post-processing step is to use stemming to evaluate the keywords. Word stemming reduces the word to its most basic format and is used across many applications of natural language processing studies [2, 39]. With stemming, both the author-assigned keywords and the extracted keywords are post-processed using the Porter stemmer [33] in the NLTK python package [26].

Exact match serves as a lower bound on the model performance as partial matches are considered incorrect. For example, if the phrase is “cancer therapy", a model that identifies “cancer" will obtain the same score as another model that fails to identify “cancer". Thus we also evaluate the algorithms using partial matching, an alternative performance measure. The partial matching score is assessed by measuring the number of matching tokens in the extracted keyphrases with ground truth keyphrases [32]. For an extracted keyphrase e and a set of ground truth keyphrase $Truth = \{t_1, t_2, \cdots, t_n\}$, the (partial match) score for e is calculated with the below formula:

$$score(e) = \underset{t \in Truth}{\operatorname{argmax}} \frac{2 \cdot common(e,t)}{|e|_T + |t|_T}, \qquad (1)$$

where $common(e,t)$ is the number of common tokens between e and t. The operator $|x|_T$ is the number of tokens in phrase x. Thus, instead of a binary score, partial matching gives credit for matching tokens for an extracted keyphrase e. $score(e)$ yields a floating point number between 0 and 1, with 1 denoting an exact match.

3.2 Extractive Keyphrase Evaluation

Supervised extractive keyphrase models often require a significant number of samples and can be quite computationally expensive to train. Moreover, many supervised keyphrase extraction algorithms compare to their unsupervised counterparts. Thus, we focus on benchmarking the unsupervised models available in the open-source python-based keyphrase extraction toolkit [8]. The `pke` module contains both graph-based models (e.g.,TextRank [30], SingleRank [43], TopicRank [10], TopicalPageRank [40], PositionRank [16], and MultipartiteRank [9]) and statistical models (e.g., Tfidf [34], YAKE [11] and KPMiner [5]).

For evaluation purposes, we select Tfidf, YAKE, KPMiner, TextRank, SingleRank, TopicalPageRank, PositionRank, and MultipartiteRank and compare the results in the extractive keywords (i.e., keywords inside the abstract). Table 2 summarizes the results on the test dataset for the 8 unsupervised algorithms. As can be seen, the precision, recall, and F1 score for all the methods are extremely low across the top 5, 10, and 15 extracted keywords. Extracting more keywords decreases the precision while increasing the recall. However, improving recall does not always yield a better F1 score.

The second and third set of columns in Table 2 illustrate the impact of stemming and partial matching with stemming, respectively. Stemming and partial matching both provide a noticeable boost in performance. Although the trends in precision, recall, and F1 score remain the same (e.g., extracting more keyphrases does not yield better F1 scores) across the three evaluation measures, it is important to note that results are substantially higher under partial matching with approximately a two-fold increase. The results also demonstrate the difficulty associated with biomedical keyword extraction as the F1 score solely on the extractive dataset is the same as other work that considers the entire gold standard (i.e. abstractive and extractive keywords) [17].

3.3 Abstractive Keyphrase Extraction

While existing studies have focused predominantly on extractive keyphrase extraction for PubMed, *PubMedAKE* also provides abstractive keyphrases from PubMed articles. Abstractive keyphrase extraction focuses on generating unseen keyphrases with a given abstract, which is a form of text generation. Many studies use sequence to sequence with encoder-decoder architecture for keyphrase generation [29], and then enforce the generated keyphrases to be based on the document topic [44]. Since we were unable to reproduce

[2] https://doi.org/10.5281/zenodo.6330817
[3] https://github.com/GarfieldLeo/PubMedAKE

		Exact matching			**Exact matching w/ stems**			**Partial matching w/ stems**		
Method		**@5**	**@10**	**@15**	**@5**	**@10**	**@15**	**@5**	**@10**	**@15**
Tfidf	**P**	0.0544	0.0409	0.0328	0.1253	0.0923	0.0737	0.1817	0.1256	0.0976
	R	0.0856	0.1290	0.1551	0.1974	0.2909	0.3484	0.2862	0.3957	0.4614
	F1	0.0665	0.0621	0.0541	0.1533	0.1402	0.1217	0.2222	0.1907	0.1612
KPMiner	**P**	0.0761	0.07338	0.0736	0.1936	0.1865	0.1857	0.3052	0.2916	0.2903
	R	0.0311	0.0320	0.0320	0.0793	0.0808	0.0809	0.1250	0.1264	0.1265
	F1	0.0442	0.0446	0.0447	0.1125	0.1128	0.1127	0.1774	0.1764	0.1762
Yake	**P**	0.0206	0.0185	0.0168	0.0889	0.0787	0.0703	0.1622	0.1235	0.1021
	R	0.0579	0.1039	0.1413	0.1480	0.2619	0.3512	0.2547	0.3879	0.4809
	F1	0.0305	0.0315	0.0301	0.1110	0.1209	0.1172	0.1982	0.1974	0.1684
TextRank	**P**	0.0123	0.0112	0.0111	0.0458	0.0457	0.0456	0.1455	0.1105	0.0951
	R	0.0343	0.0663	0.0933	0.0761	0.1483	0.2113	0.2416	0.3585	0.4406
	F1	0.0181	0.0201	0.0199	0.0572	0.0699	0.0751	0.1816	0.1689	0.1564
SingleRank	**P**	0.0172	0.0169	0.0160	0.0654	0.0653	0.0622	0.1610	0.1213	0.1014
	R	0.0481	0.0945	0.1345	0.1089	0.2173	0.3098	0.2682	0.4040	0.5051
	F1	0.0253	0.0286	0.0286	0.0817	0.1004	0.1036	0.2012	0.1866	0.1689
TopicalRank	**P**	0.0303	0.0219	0.0173	0.1258	0.0891	0.0702	0.2090	0.1446	0.1125
	R	0.0848	0.1223	0.1457	0.2095	0.2961	0.3478	0.3480	0.4810	0.5578
	F1	0.0446	0.0370	0.0310	0.1572	0.1369	0.1168	0.2612	0.2224	0.1872
PositionRank	**P**	0.0268	0.0223	0.0194	0.1027	0.0859	0.0753	0.1834	0.1336	0.1091
	R	0.0751	0.1247	0.1630	0.1712	0.2859	0.3744	0.3055	0.4446	0.5427
	F1	0.0395	0.0378	0.0347	0.1284	0.1321	0.1254	0.2292	0.2054	0.1818
MultipartiteRank	**P**	0.0322	0.0241	0.0198	0.1336	0.0983	0.0798	0.2154	0.1496	0.1179
	R	0.0902	0.1353	0.1665	0.2224	0.3270	0.3966	0.3587	0.4977	0.5819
	F1	0.0475	0.0410	0.0354	0.1669	0.1512	0.1329	0.2691	0.2301	0.1949

Table 2: Precision (P), recall (R), and F1 score (F1) for the baseline unsupervised methods at 5, 10, 15 keywords extracted with exact matching, stemming, and partial matching with stems.

	Exact matching w/ stems			**Partial matching w/ stems**		
	@5	**@10**	**@15**	**@5**	**@10**	**@15**
P	0.0143	0.0102	0.0034	0.0209	0.0196	0.0156
R	0.0120	0.0293	0.0571	0.0238	0.0473	0.0502
F1	0.0131	0.0151	0.0064	0.0223	0.0277	0.0238

Table 3: Precision (P), recall (R), and F1 score (F1) for the baseline abstractive keyphrase extraction methods with 5, 10, 15 number of keywords extracted with stemming and partial matching with stems.

various abstractive keyphrase extraction algorithms, we only implemented a simple baseline method to showcase the use of abstractive keyphrases in *PubMedAKE*.

The baseline abstractive method first creates a summarized version of the article and then uses the unsupervised keyphrase extraction algorithms to extract keyphrases. We used the built-in summarizer from HuggingFace's Transformers package [45], an open-source Python library that contains state-of-the-art natural language processing models, to summarize the title and abstract using 50 to 200 words. The summarizer is a generative summarizer, which means it creates new sentences and words from the input text. After obtaining the generated text summary, the text is put into the MultipartiteRank algorithm to extract keyphrases. We note that MultipartiteRank has the highest F1 score in the extractive baseline evaluation (see Table 2).

The performance of the baseline abstractive method yields poor results as shown in Table 3. There is almost a 10-fold decrease in performance from extractive to abstractive for the same unsupervised algorithm. This suggests that the abstractive keyphrase is an extremely difficult task. We hypothesize that to obtain better results, algorithms need to be trained on biomedical-specific data as the HuggingFace transformer model is trained on general domain text. As shown in previous studies, general state-of-the-art models often do not transfer well to biomedical text mining tasks [18, 19, 20, 23, 25]. By releasing *PubMedAKE*, future studies can develop biomedical-specific abstractive keyphrase algorithms as training data is abundant.

4 CONCLUSION AND FUTURE WORK

Keywords extraction is an ever-growing research area, and it is an especially hard task to perform on biomedical articles. As noted by previous studies, named entities, nouns, and noun phrases are peculiar and hard to identify [18, 19, 20, 24]. We constructed *PubMedAKE*, the largest keyword dataset, using all the non-commercial use articles in the PubMed Open Access Subset. The experiments demonstrate that existing state-of-the-art algorithms fail to match their performance on *PubMedAKE* when compared to general domain literature. The hope is to facilitate further research not only in biomedical literature but keyword extraction algorithms.

The experimental results also highlight several areas for future work. The evaluation metric is one direction that needs considerable attention. Even with word stemming and partial matching, precision, recall, and F1 score only focuses on keywords themselves instead of the meanings of keywords. This is important as identifying the keywords “high blood pressure" should be considered a match with “hypertension" as they convey similar meanings. Tuning the extraction algorithms for biomedical-specific nomenclature is also essential. For example, BioBERT can improve the extraction results but may require further extensions to achieve comparable performance to the general domain. Moreover, summarizing biomedical articles can be considerably different than the general domain as articles often have predefined abstract structures (e.g., Introduction, Methods, Results, Conclusions).

ACKNOWLEDGEMENTS

This work was funded by NSF grants IIS-1838200, IIS-2145411, CNS-2124104, and NIH grants 5R01LM013323-03, 5K01LM012924-03.

REFERENCES

[1] Titipat Achakulvisut, Daniel Acuna, and Konrad Kording. 2020. Pubmed parser: a python parser for pubmed open-access xml subset and medline xml dataset xml dataset. *Journal of Open Source Software*, 5, 46, 1979.

[2] Shabbir Ahmed and Farzana Mithun. 2004. Word stemming to enhance spam filtering. In *CEAS*.

[3] 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*, 2551–2557.

[4] M A Andrade and A Valencia. 1998. Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families. *Bioinformatics*, 14, 7, 600–607.

[5] Samhaa R. El-Beltagy and Ahmed Rafea. 2010. KP-miner: participation in SemEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, 190–193.

[6] D Blake. 1994. Indexing the british heart journal: choice of keywords. *British heart journal*, 71, 3, 212.

[7] Seong-Yong Bong and Kyu-Baek Hwang. 2011. Keyphrase extraction in biomedical publications using mesh and intraphrase word co-occurrence information. In *Proceedings of the ACM fifth international workshop on Data and text mining in biomedical informatics*, 63–66.

[8] Florian Boudin. 2016. Pke: an open source python-based keyphrase extraction toolkit. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, 69–73.

[9] 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)*, 667–672.

[10] Adrien Bougouin, 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*, 543–551.

[11] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509, 257–289.

[12] Kathi Canese and Sarah Weis. 2013. Pubmed: the bibliographic database. *The NCBI Handbook*, 2, 1.

[13] 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)*, 1435–1446.

[14] 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*, 4057–4066.

[15] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734.

[16] Corina Florescu and Cornelia Caragea. 2017. PositionRank: 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)*, 1105–1115.

[17] Ygor Gallina, Florian Boudin, and Béatrice Daille. 2020. Large-scale evaluation of keyphrase extraction models. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, 271–278.

[18] Zelalem Gero and Joyce Ho. 2021. Word centrality constrained representation for keyphrase extraction. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, 155–161.

[19] Zelalem Gero and Joyce C Ho. 2021. Uncertainty-based self-training for biomedical keyphrase extraction. In *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*. IEEE, 1–4.

[20] Zelalem Gero and Joyce C. Ho. 2019. Namedkeys: unsupervised keyphrase extraction for biomedical documents. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics* (BCB '19). Niagara Falls, NY, USA, 328–337.

[21] Isidoro Gil-Leiva and Adolfo Alonso-Arroyo. 2007. Keywords given by authors of scientific articles in database descriptors. *Journal of the American society for information science and technology*, 58, 8, 1175–1187.

[22] Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, 216–223.

[23] Zhenchao Jiang, Lishuang Li, Degen Huang, and Liuke Jin. 2015. Training word embeddings for deep learning in biomedical text mining tasks. In *2015 IEEE international conference on bioinformatics and biomedicine (BIBM)*. IEEE, 625–628.

[24] Sun Kim, Lana Yeganova, Donald C Comeau, W John Wilbur, and Zhiyong Lu. 2018. Pubmed phrases, an open set of coherent phrases for searching biomedical literature. *Scientific data*, 5, 1, 1–11.

[25] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36, 4, 1234–1240.

[26] Edward Loper and Steven Bird. 2002. NLTK: the natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, 63–70.

[27] Vanessa Loza, Shibamouli Lahiri, Rada Mihalcea, and Po-Hsiang Lai. 2014. Building a dataset for summarization and keyword extraction from emails. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2441–2446.

[28] 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)*, 582–592.

[29] 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)*, 582–592.

[30] Rada Mihalcea and Paul Tarau. 2004. TextRank: bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 404–411.

[31] Justin Samuel Payan. 2018. *Keyphrase extraction from scientific literature using joint geometric graph embedding matching*. Ph.D. Dissertation. University of Georgia.

[32] Jakub Piskorski, Nicolas Stefanovitch, Guillaume Jacquet, and Aldo Podavini. 2021. Exploring linguistically-lightweight keyword extraction techniques for indexing news articles in a multilingual set-up. In *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, 35–44.

[33] Martin F. Porter. 2001. Snowball: a language for stemming algorithms. (2001).

[34] Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning* number 1. Vol. 242, 29–48.

[35] Dhruva Sahrawat et al. 2020. Keyphrase extraction as sequence labeling using contextualized embeddings. *Advances in Information Retrieval*, 12036, 328.

[36] Tokala Yaswanth Sri Sai Santosh, Debarshi Kumar Sanyal, Plaban Kumar Bhowmick, and Partha Pratim Das. 2020. Dake: document-level attention for keyphrase extraction. *Advances in Information Retrieval*, 12036, 392.

[37] Kamal Sarkar. 2013. A hybrid approach to extract keyphrases from medical documents. *International Journal of Computer Applications*, 63, 18, 14–19.

[38] Alexander Thorsten Schutz et al. 2008. Keyphrase extraction from single documents in the open domain exploiting linguistic and statistical methods. *M. App. Sc Thesis*.

[39] Jasmeet Singh and Vishal Gupta. 2016. Text stemming: approaches, applications, and challenges. *ACM Comput. Surv.*, 49, 3, Article 45, 46 pages.

[40] 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* (WWW '15 Companion). Florence, Italy, 121–122.

[41] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 3104–3112.

[42] Siw Waffenschmidt, Marco Knelangen, Wiebke Sieben, Stefanie Bühn, and Dawid Pieper. 2019. Single screening versus conventional double screening for study selection in systematic reviews: a methodological systematic review. *BMC Medical Research Methodology*, 19, 132.

[43] Xiaojun Wan and Jianguo Xiao. 2008. CollabRank: towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, 969–976.

[44] Yue Wang, Jing Li, Hou Pong Chan, Irwin King, Michael R. Lyu, and Shuming Shi. 2019. Topic-aware neural keyphrase generation for social media language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2516–2526.

[45] Thomas Wolf et al. 2020. Transformers: state-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45.