Automatic Keyword Extraction: a literature review

Matteo Mortella
Independent Researcher
matteo.mortella@gmail.com

Abstract

Automatic Keyword Extraction (AKE) is a fundamental Natural Language Processing (NLP) task that plays a critical role in various applications, including information retrieval, document summarization, and content categorization. The review's purpose is to gather all the techniques, methodologies, and advancements available in the literature, and present them to researchers, practitioners, and developers interested in the topic, who can use this work as a starting point for their research, or to have a quick insight on what are the main trend in the automatic keyword extraction task. The review provides an in-depth analysis of traditional approaches, supervised and unsupervised, as well as emerging techniques that employ neural networks and deep learning architectures like transformers. This work also provides information regarding existing datasets and benchmarks and how to obtain them, as well as functioning code for building practical applications. At the end of the paper, there is a description of the state-of-the-art for automatic keyterm extraction, paired with a discussion on the evaluation metrics.

Keywords: automatic keywords extraction, key terms, key phrases, natural language processing, machine learning, deep learning, supervised learning, unsupervised learning, transformers.

1 Task description/Problem statement

The addressed NLP task is automatic keyword extraction (AKE), also known as terminology mining or term extraction, it is opposed to manual keyword extraction (MKE), in the future AKE aims to replace MKE [1]. For the rest of this document, ”keyword extraction”, ”term extraction”, ”keyphrase extraction”, ”automatic keyword extraction”, ”AKE”, ”automatic term extraction” and ”ATE” are going to be used as synonyms. The general goal of AKE is to identify relevant terms from text that can be either single or groups of words; These terms are often specialized words or phrases that carry specific meanings within a particular subject area, industry, or domain. The extracted terms can help to improve other NLP tasks such as information retrieval, topic modeling, and machine translation [2]. The problem is complex because keywords' importance and patterns vary in different contexts and domains, so a general and adaptable algorithm has to be found for accurate and meaningful extraction. Moreover, even the human-assigned keywords can vary significantly from person to person, demonstrating the subjective and articulate nature of this task.

1.1 Examples

Example table:

Input Text	**Output terms**
Climate change refers to long-term shifts in temperatures and weather patterns. Such shifts can be natural, due to changes in the sun's activity or large volcanic eruptions	climate change, weather patterns, such shifts, shifts, temperatures, changes, sun, activity, eruptions
Wine is an alcoholic drink typically made from fermented grapes. Yeast consumes the sugar in the grapes and converts it to ethanol and carbon dioxide, releasing heat in the process.	alcoholic drink, grapes Yeast, carbon dioxide

The table presents two examples of expected input and output for ATE task, the input has been found online through web search, while the output has been generated by two online tools:

the first by FiveFilters term extractor, the second by TerMine. Both tools will be described in the related section.

1.2 Real-world applications

There are several real-world applications of AKE, some of which are listed and briefly described:

- Contextual advertising: users of online platforms receive personalized ads based on their content or behavior (e.g. social networks), ATE provides useful inputs for advertising algorithms improving quality and return on advertising. [3, 4, 5]
- Valid aid for summarization: through the terms extracted from large and numerous texts is possible, in little time, to have precious insights into the content that would otherwise require minutes or hours of reading. [6, 7]
- Ontology learning: an ontology is a structured representation of knowledge that defines concepts, their properties, and the relationships between them in a particular domain. Since manual building of a domain ontology is often inadequate for new applications, ATE can be a possible method to build or enrich existing ontologies. [8]

2 Related work

After many years of research, several methods and tools have been developed for the automatic extraction of keywords. Among the available literature is possible to identify two principal choices that have an important impact on the output of ATE: feature selection and extraction method.

2.1 Features

Feature selection regards the choice of which data can help establish the relevance of terms within a document, or in other words the ”keyness” of terms, an indicator that highlights whether a word is a keyword in a domain or not comparing its general and in-domain frequency. [9, 10]
Some authors claim that morphological and syntactical features can help to extract keywords in text, in particular, some parts of speech (nouns and adjectives) have been recognized as more likely to appear as keywords. [11, 12] The following features aim to generate a list of candidate keywords from the corpus.

Term frequency-inverse document frequency
Term frequency–inverse document frequency, also referred to as TF-IDF (read this page for further information), is the most widely used statistical feature for ATE, often with some variations. This feature allows to capture the specificity of words in certain documents.

Word co-occurrence
Another statistical feature that has been used in many research is word co-occurrence. The idea behind it is to relate terms that appear together within a context window. There are contrasting results to the fact that this feature alone outperforms TF-IDF. [13, 14] Some researchers have shown that short context windows perform better than longer ones, due to the fact that long windows fail to capture the words relation effectively.

Word or sentence similarity
Other studies have questioned word or sentence similarity as a potential feature for AKE. The aim is to group words or sentences together on the basis of their meanings and senses; An operation which has been performed using diverse methods over the years: WordNet [15], bag of words (BOW), Word2Vec [16], etc. Two popular measures for similarity are the cosine and the Jaccard ones. According to the literature, word similarity is a valid feature for the AKE task since the implementations of it outperform baseline models' precision, and can help to reduce the set of candidate keywords by removing redundant terms. [17, 18, 19, 20]

Structural and positional features
Several authors have shed light on the importance of structural and positional features, for example, capitalized words, terms within quotation marks, and expressions in italic or bold font are more likely to be relevant in their context. It has been noted also that if the documents present a precise structure (i.e. scientific articles) some areas of the text are more informative than others.[4, 21, 22]

2.2 Extraction methods

Extraction methods can be divided into supervised, unsupervised, and deep learning-based techniques that try to either classify or rank the candidate keywords selected on the basis of the chosen features and produce the output keywords.

Supervised approaches
These methods were the first to be studied in the late '90s and early '00s. They treat ATE as a classification problem and so train a classifier on the features described above. In general, supervised approaches perform well in extracting relevant terms, but having the right amount of training data can be challenging since they have to be generated manually.
Popular sources of training data for ATE are academic papers or news articles since the authors tend to write down key phrases to facilitate web indexing or search; Is to be noted that sometimes authors do not choose keywords that best describe the content, but they may select words that maximize the likelihood of being noticed by searchers. Moreover, these words are subjective and a different person could choose completely different words or even terms that do not appear explicitly in the text. These complications have an impact on the training of supervised models. [23]
Some authors have noted that the average length of the documents plays a role in generalization, therefore could be useful to have different classifiers on the basis of document length.
During the course of the years, many techniques have been proposed, but the vast majority of the approaches follow these standard steps, which may vary slightly from author to author [24, 25, 26, 27, 28]:

- A preliminary phase of cleaning and sentence selection is performed, the sentences are chosen or discarded based on: fixed maximum lengths (typically unigrams, bigrams, and trigrams), the presence of some parts of speech tags, punctuation, and stop-words (articles, prepositions, conjunction, etc.).
- terms of the selected sentences are case-folded and stemmed through various algorithms.
- The obtained stemmed sentences are then scored and ranked by some measures based on frequency, position, or part of speech tags; Popular ones are TF-IDF and the position of the first occurrence of each word in the document. Also, sentences with nouns and adjectives get a better score. Some authors exploited domain-specific lists of words to directly spot keywords.
- a fixed number of the ranked sentences is extracted and for them, some other features for classification purposes are computed, be they statistical or morpho-syntactical based.
- a classifier (popular ones are Naive Bayes, Support Vector Machines, and decision trees), is trained on a training set that could be balanced with several techniques, and evaluation is then performed by comparing the number of matches between true human-assigned and automatically extracted keywords.

Unsupervised approaches
To overcome the problem of finding a decent amount of training data, unsupervised approaches have been studied, starting from the early 2000s until the present day. They treat AKE as a ranking or clustering problem.
Most of the researchers built graphs on the documents to perform the keyword extraction task [29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39], while a few others gave importance to the spatial distribution of words and experimented with structural, frequency-based, and co-occurrence features [40, 41, 42, 43]; Is to be noted that some of these authors put the attention on a feature that has not been sufficiently explored for ATE before: information entropy. It comes from Shannon's theory of information, and its utility in keyphrases extraction has been proven: relevant terms for a document tend to concentrate in specific areas and this allows to cluster and identify them.
According to the literature, is possible to define some standard steps that an unsupervised approach follows:

- Text preprocessing.
  This step is common to both graph-based and non-graph-based approaches. It consists of tokenization, treatment of stop-words, text annotation, and eventual application of custom rules based on lexical, morphological, and/or linguistic criteria.
- Graph/features building.
  Defining what are the vertexes and the edges is crucial; Words, sentences, and even entire topics (identified through latent Dirichlet allocation) have been regarded as nodes. Edges

represent the relationships between vertexes, typically co-occurrent entities within a window have an edge that could be oriented or not depending on the envisaged strategy. For non-graph approaches, the features for each candidate keyword are computed.

- Scoring phase.
  The vertexes or the candidate keywords are scored according to some criteria. In graph-based studies, centrality measures like position, degree, or closeness between vertexes have been used to weigh their importance, in some cases, scores based on linguistics or Word2Vec's similarity have been used (e.g. semantic relatedness). In the other kinds of studies, position-based and frequency-based measures were proposed.

- Sorting and ranking.
  In this phase, the candidate keywords and nodes, that now have scores, are sorted, and the top N of them is regarded as a keyword. N can impact seriously the final results so the researchers tend to experiment with several values of it.

- Evaluation.
  The methods are tested on the datasets used for the supervised approaches and the automatically extracted keywords are compared with the real ones. Precision, Recall, and F1-score are common measures.

Transformers and attention-based models
These are the most recent, both supervised and unsupervised, approaches that are being studied for addressing automatic keyword extraction. [44, 45, 46, 47]
The research is still immature, in fact, the first pre-trained models for this task were released in 2020. The few available studies exploited pre-trained models like BERT [48], KeyBert [49] and T5 Text-to-text [50].

3 Datasets and benchmarks

There are many datasets available for automatic keyword extraction, most of them come from the academic world, are solely in English, and have a focus on a few subjects. The datasets are described with their benchmarks below. At the end of this section, there is a summarizing table.
All the metrics are computed on a number of automatically extracted keyphrases equal to ten.

- Inspec [51]
  It consists of 2,000 abstracts of scientific journal papers in computer science collected between the years 1998 and 2002. Each abstract has two sets of keyphrases annotated by professional indexers - controlled and uncontrolled. The uncontrolled keyphrases are those selected by the indexers after reading the full-length scientific articles. The controlled keyphrases are obtained from the Inspec thesaurus and therefore are often not present in the abstract's text.
  The benchmark for this dataset is represented by Phraseformer [45], an unsupervised transformer-based model that performed a 71.67 score for F-1.

- Semeval2010 [52]
  The dataset consists of 284 scientific articles in the computer science domain with keyphrases carefully chosen by both their authors and readers. The papers have a length between 6 to 8 pages. They treat the following computer science research areas: distributed systems, information search, and retrieval.
  The benchmark is 48.65 for F-1 score, performed by Phraseformer.

- Semeval2017 [53]
  This dataset is a new version of the Semeval2010, it is a collection of 500 paragraphs selected from 500 ScienceDirect journal articles, that treat topics of computer science, material sciences, and physics. Each document has keywords selected by an annotator. With a score of 67.13 for F-1 score, Phraseformer is the benchmark for this dataset.

- Krapivin [54]
  The dataset has high quality and consists of 2.304 academic papers from the Computer Science domain published by ACM. Each paper has its keyphrases assigned by the authors and verified by the reviewers. Different parts of papers, such as the title and abstract, are separated.
  The benchmark for F-1 score is 16.71,

reached by PromptRank [46] a transformer-based model.

- KP20k [55]
  KP20k is a large-scale academic articles dataset with 528.000 articles for training, 20.000 articles for validation and 20.000 articles for testing. The benchmark is 19.2 on F-1 score, obtained by a supervised transformer-based model called UCPhrase [47].

- KPTimes [56]
  KPTimes is a large-scale dataset of news texts paired with editor-curated keyphrases. UCPhrase presents a benchmark with an F-1 score of 10.9 for KPTimes.

- NUS [57]
  NUS is a collection of 211 documents in plain text format, they are scientific conference papers, with a length range of 4-12 pages. A manual keyphrase assignment has been performed on the texts. With an F-1 score of 20.13, PromtRank is the benchmark.

To summarize the results the following table is presented:

Dataset	**Benchmark**	**F1 score**
Inspec	Phraseformer	71.67
Semeval2010	Phraseformer	48.65
Semeval2017	Phraseformer	67.13
Kaprivin	PromptRank	16.71
KP20k	UCPhrase	19.20
KPTimes	UCPhrase	10.90
NUS	PromptRank	20.13

Some of these datasets and a few others can be dowloaded at this github page.

4 Existing tools, libraries, papers with code

Regarding ATE tools, in particular online ones, two of them are presented:

- FiveFilters term extractor [58, 59]
  That is a simple and free keyword extractor, based on a first phase of pre-processing, filtering, and POS tagging to identify candidate keywords in the text, followed by a selection of them performed on the basis of statistical and linguistical criteria. The tool allows one to choose the maximum number of extracted key terms and words per keyphrase and the output format.

- TerMine [60, 61]
  Another tool available for free is TerMine, it relies on a particular measure called C-value to asses the termhood of words and so select keywords. C-value is a combination of linguistic and statistical information from the text. [62]

Many other free tools for ATE are available at this page.

Existing libraries (Python):

- Python Keyphrase Extraction [63]
  is a collection of popular keyword extraction algorithms proposed in the literature. Among them TextRank, TopicRank, and YAKE! are available.

- rake-nltk [64]
  Rapid Automatic Keyword Extraction (RAKE [37]), is a domain-independent keyword extraction graph-based algorithm that tries to determine key phrases in the text by analyzing the frequency of word appearance and its co-occurrence with other expressions in the text.

- PyTextRank [65]
  a Python implementation of TextRank and some of its variations for phrase extraction and summarization of text documents. It is part of the SpaCy [66], an open-source software library for advanced natural language processing, written in the programming languages Python and Cython.

- KeyBert [67]
  It is a minimal and easy-to-use keyword extraction model that leverages BERT embeddings to create keywords and keyphrases that are most similar to a document.

Papers with code:

- FRAKE [39, 68]
  FRAKE that stands for ”Fusional Real-time Automatic Keyword Extraction”, present code for text pre-processing, graph building, and feature extraction and computing.

- UCPhrase [47, 69]
  UCPhrase that stands for "Unsupervised Context-aware Quality Phrase Tagging", present code to train a supervised Convolutional Neural Network (CNN) to capture inter-word relationships and context information used for automatic keyterm extraction.

- WordTopic-MultiRank [35]
  WordTopic-MultiRank presents the algorithm of a novel ranking algorithm: Biased-MultiRank, which scores words and topics simultaneously since they are considered to have mutual influence on each other.

- YAKE! [43, 70]
  YAKE! which stands for "Yet Another Keyword Extractor" is an unsupervised feature-based approach, it shows the code for all the pre-processing, feature extraction, n-gram generation, keyword selection, and ranking operations.

- PromptRank [46, 71]
  PromptRank is a pre-trained language model that addresses AKE with an unsupervised approach. The code presents an implementation of PromptRank using the NLTK [72] (Natural Language Toolkit) Python's library and the T5Tokenizer.

5 State-of-the-art evaluation

In this section, a brief explanation of the metrics used for evaluation is conducted, followed by a description of the state-of-the-art for AKE. The evaluation for the addressed task relies on the comparison between automatically extracted keywords and manually annotated keywords chosen by a human being.
Typically, two keywords match when their sequence is a perfect match. For this kind of comparison, the preferred metrics are precision, recall, and F-1 score.
Particular attention has to be put to the number of keywords for each document: it has to be the same or at least similar both for the automatic and the hand-chosen keyphrases. Otherwise, in the case of more keywords identified by the algorithms, the precision will be penalized for sure, in fact even in the case of all perfect matches, the exceeding extracted keywords will always be regarded as false positives. Another aspect to consider is the possible presence of out-of-text hand-noted keywords, these should be avoided when evaluating models that are based solely on the analysis of the text. Otherwise, the recall will be lowered for sure, in fact, the lower bound on the number of false negatives will correspond to the number of out-of-text keywords.
So, the precision, recall and F-1 score metrics are regarded as P@K, R@K and F1@K where K is the number of keywords considered for the matches. Typical values for K are 3, 5, and 10. All the scores in this review are presented for K equal to 10 when possible.
A summarizing table on evaluation metrics is shown at the end of the section.
Now, some state-of-the-art models, divided by kind of method are briefly presented:

Unsepervised graph-based

- TextRank(2004) [33], even if its results have been outperformed by other models, this method deserves a special mention because it was the first graph-based approach for ATE and has laid the foundation for other subsequent research.
  It was inspired by PageRank [73], the algorithm that Google used to rank web pages at its beginning.
  The built graph have words or sentences as vertex and co-occurrence edges. The vertexes are scored by a "voting" or "recommendation" system: when one vertex links to another one, it is voting for that node. The more votes a node has the more importance it has in the text context and so the higher the score. Scored nodes are then sorted and top-K are selected as keywords. The best results that were obtained on the Inspec dataset are 36.2 of F1 score, 31.2 of precision, and 43.1 of recall. These were obtained with a sliding window of dimension 2 and a variable number of extracted keywords based on text length (one-third of the candidate keywords).

- FRAKE(2021) is the best graph-based model in terms of performance. It combines the classic centrality measures for graphs (degree, betweenness, eigenvector, closeness centrality) and textural features (casing, term position, term frequency normalization, term different sentence, part of speech tagging) to extract keywords. The model has been tested on 7 famous datasets for keyword extraction, for brevity only Inspec result is reported: F1 score on 58.9, precision of 57.2 and recall of 60.7. The number of extracted keywords is 10. This was the best result available until the advent of transformed-based models.

Unsupervised feature-based
YAKE!(2019) is the top available model in performance for the unsupervised feature-based category. It follows the classic steps for unsupervised ATE, pre-processing, feature creation, terms scoring, n-gram creation for keywords with more than one word, and ranking. The features are based on casing, normalized frequency, frequency in sentences, and position. With a fixed number of 10 keywords extracted, it has been compared with many baseline models on several datasets. The F1 score on Inspec is 31.6, as a comparison the same metric of TextRank (which results presented before are not under the constraint of a fixed number of extracted keywords) is 9.8. Precision and recall are not reported for Inspec.

Supervised methods
With Improved Automatic Keyword Extraction (2003) [28] Hulth shared with the world the Inspec dataset. Several approaches were tested, but in this section, only the best one is presented. The method considers uni-grams, bi-grams, and tri-grams as candidate keywords, then filters them with customized rules (regarding stemming, stop-words, presence of numeric characters, etc.) and finally builds features based on frequency and position for them. The machine learning model used for the supervised task is a decision tree classifier, the presented results were computed with a variable number of keywords extracted (with an average of 4.37 per document). An F1 score of 33.9 was obtained with 25.2 for precision and 51.7 for recall.

Transformer-based methods
Phraseformer [45] is a model that uses transformer and graph embedding techniques to identify candidate keywords. Then the keyphrase extraction is treated as a sequence labeling problem solved by classification. A text representation is learned through the BERT transformer and the self-attention [74] mechanism, and then the structure and context of the text are learned through a co-occurrence graph built via the ExEm embedding algorithm [75](others were studied, but here is reported only the best algorithm). Then both representations are combined (summed) into a single vector for each word. Finally, each word is labeled via a random forest to classify the word as a keyword or not. The F1 score measured on the Inspec dataset is equal to 71.67, which is the best result currently available. Precision and recall were not reported in the study.

The table below summarizes the results for the Inspec dataset, bolded values are the best result (available on the official research):

Model	F1 score	Precision	Recall
TextRank*	36 (9.8)	31.2	43.1
FRAKE	58.9	**57.2**	**60.7**
YAKE!	31.6	-	-
Hulth AKE*	33.9	25.2	51.7
Phraseformer	**71.6**	-	-

*TextRank and Hulth AKE presented results without the constraints on the number of automatically extracted keywords, when available for K = 10 the result is reported in round brackets.

6 Conclusions

This review of the literature sheds light on trends and methods that emerged over the years to solve automatic keyword extraction from text.
The first approaches were supervised and exploited simple statistical or linguistic features combined with machine learning classifiers, they laid the basis for future works.
Then due to the limited available data, the unsupervised approaches took their spot. Most of them outperformed the supervised approaches thanks to graph-based models introduced by TextRank and their capability of capturing the context through the co-occurrence of words and centrality measures. The state-of-the-art at the time has been reached by FRAKE a model that combined both graph, statistical, and linguistical features. In the past 2 years a new tendency has emerged: thanks to the introduction of the self-attention mechanism and encoder-decoder transformer architectures, large language models have been born and thanks to their astounding capability of understanding the context in the text have been applied to the AKE task. Phraseformer, a model that combines transformer-produced and graph-based features is the best model available for solving the

automatic keyword extraction task.

Analyzing the recent trends in natural language processing, and the recent proposed strategies for AKE in particular, is clear that the deep learning models are assuming a crucial role in research and studies; Future work involving these models is necessary to explore the possibility and the limitations of them for the automatic keyword extraction task.

References

[1] Kris Heylen and Dirk De Hertog. Automatic term extraction. In *Handbook of Terminology Online Volume 1*. John Benjamins Publishing Company, 2015. 1

[2] Hanh Thi Hong Tran, Matej Martinc, Jaya Caporusso, Antoine Doucet, and Senja Pollak. The recent advances in automatic term extraction: A survey, 2023. 1

[3] Kushal S. Dave and Vasudeva Varma. Pattern based keyword extraction for contextual advertising. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, page 1885–1888, New York, NY, USA, 2010. Association for Computing Machinery. 2

[4] Wen-tau Yih, Joshua Goodman, and Vitor R. Carvalho. Finding advertising keywords on web pages. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, page 213–222, New York, NY, USA, 2006. Association for Computing Machinery. 2

[5] Suleman khan, Qazi Mudassar Ilyas, and Waqas Anwar. Contextual advertising using keyword extraction through collocation. In *Proceedings of the 7th International Conference on Frontiers of Information Technology*, New York, NY, USA, 2009. Association for Computing Machinery. 2

[6] Vahed Qazvinian, Dragomir Radev, and Arzucan Özgür. Citation summarization through keyphrase extraction. In *Proceedings of the 23rd international conference on computational linguistics (COLING 2010)*, pages 895–903, 2010. 2

[7] Hongyuan Zha. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, page 113–120, New York, NY, USA, 2002. Association for Computing Machinery. 2

[8] Kiruparan Balachandran and Surangika Ranathunga. Domain-specific term extraction for concept identification in ontology construction. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 34–41, 2016. 2

[9] Costas Gabrielatos and Anna Marchi. Keyness: Matching metrics to definitions. 2011. Theoretical-methodological challenges in corpus approaches to discourse studies and some ways of addressing them ; Conference date: 05-11-2011. 2

[10] Costas Gabrielatos. *Keyness analysis: Nature, metrics and techniques*, pages 225–258. 2018. 2

[11] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004. 2

[12] Enes Altuncu, Jason R. C. Nurse, Yang Xu, Jie Guo, and Shujun Li. Improving performance of automatic keyword extraction (ake) methods using pos-tagging and enhanced semantic-awareness, 2022. 2

[13] Christian Wartena, Rogier Brussee, and Wout Slakhorst. Keyword extraction using word co-occurrence. In *2010 Workshops on Database and Expert Systems Applications*, pages 54–58, 2010. 2

[14] Yutaka Matsuo and Mitsuru Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13, 2003. 2

[15] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41, 1995. 2

[16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. 2

[17] Furkan Goz and Alev Mutlu. Skywords: An automatic keyword extraction system based on the skyline operator and semantic similarity. *Engineering Applications of Artificial Intelligence*, 123:106338, 2023. 2

[18] Talha Bin Sarwar Kamal Z. Zamli Rajan Jose M. Saef Ullah Miah, Junaida Sulaiman. Study of keyword extraction techniques for electric double-layer capacitor domain using text similarity indexes: An experimental analysis. 2021. 2

[19] Meng Wenchao, Liu Lianchen, and Dai Ting. A modified approach to keyword extraction based on word-similarity. In *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, volume 3, pages 388–392, 2009. 2

[20] Mohamed Haggag. Keyword extraction using semantic analysis. *International Journal of Computer Applications*, 61:1–6, 01 2013. 2

[21] Patrice Lopez and Laurent Romary. HUMB: Automatic key term extraction from scientific articles in GROBID. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 248–251, Uppsala, Sweden, July 2010. Association for Computational Linguistics. 2

[22] J. P. Herrera and P. A. Pury. Statistical keyword detection in literary corpora. *The European Physical Journal B*, 63:135–146, 2008. 2

[23] Lucas Sterckx, Cornelia Caragea, Thomas Demeester, and Chris Develder. Supervised keyphrase extraction as positive unlabeled learning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1924–1929. Association for Computational Linguistics, 2016. 3

[24] Peter D. Turney. Learning algorithms for keyphrase extraction, 2002. 3

[25] Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. Kea: Practical automatic keyphrase extraction. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, page 254–255. Association for Computing Machinery, 1999. 3

[26] Eibe Frank, Gordon Paynter, Ian Witten, Carl Gutwin, and Craig Nevill-Manning. Domain-specific keyphrase extraction. 07 1999. 3

[27] Min Song, Il-Yeol Song, and Xiaohua Hu. Kpspotter: A flexible information gain-based keyphrase extraction system. In *Proceedings of the 5th ACM International Workshop on Web Information and Data Management*, page 50–53. Association for Computing Machinery, 2003. 3

[28] Anette Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, page 216–223. Association for Computational Linguistics, 2003. 3, 7

[29] Slobodan Beliga, Ana Meštrović, and Sanda Martincic-Ipsic. An overview of graph-based keyword extraction methods and approaches. *Journal of Information and Organizational Sciences*, 39:1–20, 07 2015. 3

[30] Xiong Ao, Xin Yu, Derong Liu, and Hongkang Tian. News keywords extraction algorithm based on textrank and classified tf-idf. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pages 1364–1369, 2020. 3

[31] Suhan Pan, Zhiqiang Li, and Juan Dai. An improved textrank keywords extraction algorithm. In *Proceedings of the ACM Turing Celebration Conference - China*, New York, NY, USA, 2019. Association for Computing Machinery. 3

[32] Xiaojun Wan and Jianguo Xiao. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, page 855–860. AAAI Press, 2008. 3

[33] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, 2004. Association for Computational Linguistics. 3, 6

[34] Adrien Bougouin, Florian Boudin, and Béatrice Daille. TopicRank: Graph-based topic ranking for keyphrase extraction. In

Proceedings of the Sixth International Joint Conference on Natural Language Processing, Nagoya, Japan, 2013. Asian Federation of Natural Language Processing. 3

[35] Fan Zhang, Lian'en Huang, and Bo Peng. WordTopic-MultiRank: A new method for automatic keyphrase extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 10–18, Nagoya, Japan, 2013. Asian Federation of Natural Language Processing. 3, 6

[36] Tingting Zhang, Baozhen Lee, Qinghua Zhu, Xi Han, and Ke Chen. Document keyword extraction based on semantic hierarchical graph model. *Scientometrics*, 128(5):2623–2647, May 2023. 3

[37] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. *Automatic Keyword Extraction from Individual Documents*, chapter 1, pages 1–20. John Wiley Sons, Ltd, 2010. 3, 5

[38] Yujun Wen, Hui Yuan, and Pengzhou Zhang. Research on keyword extraction based on word2vec weighted textrank. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pages 2109–2113, 2016. 3

[39] Aidin Zehtab-Salmasi, Mohammad-Reza Feizi-Derakhshi, and Mohamad-Ali Balafar. Frake: Fusional real-time automatic keyword extraction, 2021. 3, 5

[40] C. Carretero-Campos, P. Bernaola-Galván, A.V. Coronado, and P. Carpena. Improving statistical keyword detection in short texts: Entropic and clustering approaches. *Physica A: Statistical Mechanics and its Applications*, pages 1481–1492, 2013. 3

[41] Ali Mehri and Amir H. Darooneh. The role of entropy in word ranking. *Physica A: Statistical Mechanics and its Applications*, pages 3157–3163, 2011. 3

[42] J. P. Herrera and P. A. Pury. Statistical keyword detection in literary corpora. *The European Physical Journal B*, pages 135–146, 2008. 3

[43] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289, 2019. 3, 6

[44] Piotr Pezik, Agnieszka Mikołajczyk-Bareła, Adam Wawrzyński, Bartłomiej Nitoń, and Maciej Ogrodniczuk. Keyword extraction from short texts with a text-to-text transfer transformer, 2022. 4

[45] Narjes Nikzad-Khasmakhi, Mohammad-Reza Feizi-Derakhshi, Meysam Asgari-Chenaghlu, Mohammad-Ali Balafar, Ali-Reza Feizi-Derakhshi, Taymaz Rahkar-Farshi, Majid Ramezani, Zoleikha Jahanbakhsh-Nagadeh, Elnaz Zafarani-Moattar, and Mehrdad Ranjbar-Khadivi. Phraseformer: Multimodal key-phrase extraction using transformer and graph embedding, 2021. 4, 7

[46] Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, and Xiaoyan Bai. Promptrank: Unsupervised keyphrase extraction using prompt, 2023. 4, 5, 6

[47] Xiaotao Gu, Zihan Wang, Zhenyu Bi, Yu Meng, Liyuan Liu, Jiawei Han, and Jingbo Shang. UCPhrase. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery &amp Data Mining*. ACM, 2021. 4, 5, 6

[48] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 4

[49] Maarten Grootendorst. Keybert: Minimal keyword extraction with bert., 2020. 4

[50] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 2020. 4

[51] Inspec dataset. https://paperswithcode.com/dataset/inspec. Accessed: 2023-09-14. 4

[52] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. SemEval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evalu-*

ation, pages 21–26. Association for Computational Linguistics, 2010. 4

[53] Semeval2017 dataset. https://paperswithcode.com/dataset/semeval2017. Accessed: 2023-09-14. 4

[54] Krapivin dataset. https://paperswithcode.com/dataset/krapivin. Accessed: 2023-09-14. 4

[55] Kp20k dataset. https://paperswithcode.com/dataset/kp20k. Accessed: 2023-09-14. 5

[56] Ygor Gallina, Florian Boudin, and Béatrice Daille. Kptimes: A large-scale dataset for keyphrase generation on news documents, 2019. 5

[57] Nus dataset. https://paperswithcode.com/dataset/nus. Accessed: 2023-09-14. 5

[58] Fivefilters explained. https://www.fivefilters.org/2013/term-extraction/. Accessed: 2023-09-14. 5

[59] Fivefilters tool. http://termextract.fivefilters.org/. Accessed: 2023-09-14. 5

[60] Termine explained. https://libraryguides.mcgill.ca/text-mining/TerMine. Accessed: 2023-09-14. 5

[61] Termine tool. http://www.nactem.ac.uk/software/termine/#form. Accessed: 2023-09-14. 5

[62] Katerina T. Frantzi, Sophia Ananiadou, and Jun-ichi Tsujii. The c-value/nc-value method of automatic recognition for multi-word terms. In *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, page 585–604, Berlin, Heidelberg, 1998. Springer-Verlag. 5

[63] Florian Boudin. pke: an open source python-based keyphrase extraction toolkit. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 69–73, 2016. 5

[64] Rake library. https://csurfer.github.io/rake-nltk/_build/html/index.html. Accessed: 2023-09-14. 5

[65] Paco Nathan. PyTextRank, a Python implementation of TextRank for phrase extraction and summarization of text documents, 2016. 5

[66] Matthew Honnibal and Ines Montani. spaCy: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. 2017. 5

[67] Keybert library. https://github.com/MaartenGr/KeyBERT. Accessed: 2023-09-14. 5

[68] Frake github. https://github.com/cominsys/FRAKE/tree/main. Accessed: 2023-09-14. 5

[69] Ucphrase github. https://github.com/xgeric/UCPhrase-exp. Accessed: 2023-09-14. 6

[70] Yake! github. https://github.com/LIAAD/yake. Accessed: 2023-09-14. 6

[71] Promptrank github. https://github.com/hlt-nlp/promptrank. Accessed: 2023-09-14. 6

[72] Edward Loper Bird, Steven and Ewan Klein. Natural language processing with python, 2009. 6

[73] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. In *The Web Conference*, 1999. 6

[74] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 7

[75] Narjes Nikzad-Khasmakhi, Mohammadali Balafar, M. Reza Feizi-Derakhshi, and Cina Motamed. ExEm: Expert embedding using dominating set theory with deep learning approaches. *Expert Systems with Applications*, 2021. 7