information

MDPI

Article

Improving Performance of Automatic Keyword Extraction (AKE) Methods Using PoS Tagging and Enhanced Semantic-Awareness

Enes Altuncu [1], Jason R. C. Nurse [1], Yang Xu [2], Jie Guo [2] and Shujun Li [1,*]

[1] Institute of Cyber Security for Society (iCSS) & School of Computing, University of Kent, Canterbury CT2 7NP, UK; drenesaltuncu@gmail.com (E.A.); j.r.c.nurse@kent.ac.uk (J.R.C.N.)
[2] School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China; xuyang2018@sjtu.edu.cn (Y.X.); guojie@sjtu.edu.cn (J.G.)
* Correspondence: s.j.li@kent.ac.uk

Abstract

Automatic keyword extraction (AKE) has gained more importance with the increasing amount of digital textual data that modern computing systems process. It has various applications in information retrieval (IR) and natural language processing (NLP), including text summarisation, topic analysis and document indexing. This paper proposes a *simple* but *effective* post-processing-based universal approach to improving the performance of *any* AKE methods, via an enhanced level of semantic-awareness supported by PoS tagging. To demonstrate the performance of the proposed approach, we considered word types retrieved from a PoS tagging step and two representative sources of semantic information—specialised terms defined in one or more context-dependent thesauri, and named entities in Wikipedia. The above three steps can be simply added to the end of *any* AKE methods as part of a post-processor, which simply re-evaluates all candidate keywords following some *context-specific* and *semantic-aware* criteria. For five state-of-the-art (SOTA) AKE methods, our experimental results with 17 selected datasets showed that the proposed approach improved their performances both *consistently* (up to 100% in terms of improved cases) and *significantly* (between 10.2% and 53.8%, with an average of 25.8%, in terms of F1-score and across all five methods), especially when all the three enhancement steps are used. Our results have profound implications considering the fact that our proposed approach can be easily applied to any AKE method with the standard output (candidate keywords and scores) and the ease to further extend it.

Keywords: keyword extraction; pos tagging; semantic-awareness; context-awareness

Academic Editor: Katsuhide Fujita

Received: 19 May 2025
Revised: 5 July 2025
Accepted: 9 July 2025
Published: 13 July 2025

Citation: Altuncu, E.; Nurse, J.R.C.; Xu, Y.; Guo, J.; Li, S. Improving Performance of Automatic Keyword Extraction (AKE) Methods Using PoS Tagging and Enhanced Semantic-Awareness. *Information* **2025**, *16*, 601. https://doi.org/10.3390/info16070601

1. Introduction

Keyword extraction (KE), also known as keyphrase or key term extraction, is an information extraction task that aims to identify a number of words/phrases that best summarise the nature or the context of a piece of text. It has several applications in information retrieval (IR) and natural language processing (NLP), including text summarisation, topic analysis, and document indexing [1,2]. Considering the vast amount of text-based documents online in today's digital society, it is very useful to be able to extract keywords from online documents automatically to support large-scale textual analysis. Therefore, for many years, the research community has been investigating automatic keyword extraction (AKE) methods, especially with the recent advancements in artificial intelligence (AI) and NLP. Despite these efforts, however, AKE has been shown to be a challenging

task, and AKE methods with very high performance are still to be found [3]. Two main challenges are the lack of a precise definition of the AKE task and the lack of consistent performance evaluation metrics and benchmarks [1]. Since there is no consensus on the definition and characteristics of a *keyword*, KE datasets created by researchers have different characteristics. Examples include the minimum/average/maximum numbers of keywords, if absent keywords (human-labelled keywords that do not appear in the text) are allowed, and what part-of-speech (PoS) tags, such as verbs, are accepted as valid keywords. This makes performance evaluation and comparison of AKE methods more difficult.

Based on whether a labelled training set is used, AKE methods reported in the literature can be grouped into unsupervised and supervised methods. Unsupervised methods include statistical, graph-based, embedding-based and/or language model-based methods, while supervised ones use either traditional or deep machine learning models [3]. Surprisingly, for most AKE methods, semantic information has not been considered or only insufficiently considered to align the returned keywords with the semantic context of the input document [4].

In this work, to fill the above-mentioned gap on the lack of or insufficient use of semantic information in the state-of-the-art (SOTA) AKE methods, we propose a *universal* performance improvement approach for *any* AKE methods. This approach serves as a post-processor that can consider semantic information more explicitly, with the support of PoS tagging. To start with, we conducted an analysis of human-annotated 'gold standard' keywords in 17 KE datasets to better understand some relevant characteristics of such keywords. Particularly, this analysis focuses on PoS tag patterns, *n*-gram sizes, and the possible consideration of semantic information by human labellers when extracting keywords.

Our proposed approach is demonstrated using the following three post-processing steps that can be freely combined: (1) keeping candidate keywords with a desired PoS tag only; (2) matching candidate keywords with one or more context-specific thesauri containing more semantically relevant terms; and (3) prioritising candidate keywords that appear as a valid Wikipedia named entity. We applied different combinations of the above three post-processing steps to five SOTA AKE methods, YAKE! [5], KP-Miner [6], RaKUn [7], LexRank [8], and SIFRank+ [9], and compared the performances of the original methods with those of the enhanced versions. The experimental results with the 17 KE datasets showed that our proposed post-processing steps helped improve the performances of all the five SOTA AKE methods both *consistently* (up to 100% in terms of improved cases) and *significantly* (between 10.2% and 53.8%, with an average of 25.8%, in terms of F1-score and across all five methods), particularly when all the three steps are combined. Our work validates the possibility of using easy-to-use post-processing steps to enhance the semantic awareness of AKE methods and to improve their performance in real-world applications, a fact that has not been reported before (to the best of our knowledge). The main contributions of this paper are as follows:

- We propose a modular and universal post-processing pipeline that enhances existing AKE methods using part-of-speech filtering and external knowledge sources.
- We provide a comprehensive analysis of 17 AKE datasets to empirically justify our design choices.
- We conduct extensive experiments to demonstrate that the proposed pipeline improves the performance of multiple state-of-the-art AKE methods across diverse evaluation settings.

The rest of the paper is organised as follows. Section 2 briefly surveys AKE methods in the literature. The analysis of the human-annotated keywords in 17 KE datasets is given in Section 3. In Section 4, we present the methodology of our study. Section 5 explains the

experimental setup for evaluation as well as the results. Finally, the paper is concluded with some further discussions in Section 6, and an overall summary in Section 7.

2. Related Work

2.1. Unsupervised AKE Methods

Some unsupervised AKE methods have been proposed, including statistical, graph-based, embedding-based and language model-based methods [3]. Statistical AKE methods rely on some selected statistical metrics, e.g., term frequency, relevance to context, and co-occurrences, for ranking candidate keywords. One of the most used metrics is TF-IDF [10], which combines two aspects of a term: term frequency within the input article, and the inverse document frequency across several domains. One AKE method using TF-IDF is KP-Miner [6], which also considers other metrics such as word length and word position. A more recent method in this category is YAKE! [5]. It leverages a range of statistical metrics, such as casing, word position, word frequency, word relatedness to context, and how often a term appears in different sentences. Finally, LexSpec [8] makes use of lexical specificity—a statistical metric to select the most representative keywords from a given text based on the hypergeometric distribution. While statistical AKE methods are easy to compute and language-independent, they mostly fail to capture contextual or semantic significance, leading to poor performance in nuanced texts.

Graph-based AKE methods consider candidate keywords as nodes in a directed graph, often with weighted edges reflecting the syntactic/semantic relatedness of different keywords. They leverage graph-based methods, such as PageRank [11], for ranking the nodes of the graph in terms of their overall importance. The earliest AKE method in this category is TextRank [12]. It uses an unweighted graph of candidate keywords after filtering the ones that are not nouns or adjectives, and uses PageRank for ranking the nodes. As an extension to TextRank, SingleRank [13] adds edge weights to the graph, which reflect the number of co-occurrences of the candidate keywords represented by any pair of two connected nodes. Another graph-based AKE method is RAKE [14], which builds a word-word co-occurrence graph and assigns a score for each candidate by using word frequency and word degree. A more recent graph-based AKE method is RaKUn [7], which introduces meta-vertices by aggregating similar vertices and employs load centrality metrics for candidate ranking. Finally, LexRank [8] and TFIDFRank [8] are two different enhanced versions of SingleRank, which use lexical specificity and TF-IDF, respectively. Graph-based AKE approaches improve upon the statistical approaches by modelling term connectivity, yet still depend heavily on co-occurrence patterns.

Embedding-based AKE methods utilise word representation techniques, such as Doc2Vec [15] and GloVe [16]. An example method in this category is EmbedRank [17], which uses sentence embeddings and ranks candidate keywords in terms of cosine similarity. A more recent method is SIFRank [9], which combines a sentence embedding model SIF and an autoregressive pre-trained language model ELMo, and it was upgraded to SIFRank+ by position-biased weight to improve its performance for long documents. Lastly, MDERank [18] considers the similarity between the embeddings of the source document and its masked version for candidate ranking. Embedding-based AKE methods can better capture context and semantics, but often come with a higher computational cost. Furthermore, many embedding-based approaches depend on pre-trained language models, which may require fine-tuning or adaptation to specific domains. Their reliance on large-scale models also reduces interpretability and makes integration into lightweight systems more challenging.

Apart from the AKE methods mentioned above, there also exist a number of AKE methods based on other techniques. Rabby et al. [19] proposed TeKET, a domain- and

language-independent AKE technique utilising a binary tree for extracting final keywords from candidate ones. As another example, Liu et al. [20] introduced an AKE algorithm based on term clustering considering semantic relatedness to identify the exemplar terms. The identified exemplar terms are then used to extract keywords.

2.2. Supervised AKE Methods

Although unsupervised methods are preferred for AKE, supervised methods have also been proposed. One of the earliest methods is KEA [21], which calculates TF-IDF scores and the position of the first occurrence of each candidate, and employs the Naive Bayes learning algorithm to decide if a candidate should be selected. More recently, there has been a growing interest in using deep learning for AKE. For example, Basaldella et al. [22] proposed an AKE method based on Bi-LSTM, which is capable of exploiting the context of each candidate word. Another AKE method, TNT-KID [23], leverages transformers and allows users to train their own language model on a domain-specific corpus. A third example is TANN [24], an AKE method based on a topic-based artificial neural network model. It aims to improve the performance of AKE by transferring knowledge from a resource-rich source domain to an unlabelled or insufficiently labelled target domain. Finally, Bordoloi et al. [25] proposed a supervised variant of TextRank, leveraging a statistical supervised weighting scheme for terms to employ both global and local weights during keyword extraction. Supervised AKE methods can be promising when trained on large annotated corpora. However, their reliance on labelled data limits their applicability in low-resource domains or languages, and their generalisation to unseen domains can be inconsistent. Additionally, such methods tend to require significant computational resources during both training and inference.

2.3. PoS Tagging and Semantics in AKE

Many AKE methods have considered how to extract more semantically meaningful keywords. For this purpose, PoS tagging has been used so that extracted keywords are restricted to a pre-defined set of PoS tag patterns, e.g., noun phrases only [26–28]. Some methods utilise external knowledge to provide useful contextual information for extracting more semantically sensible keywords. For instance, Li and Wang [29] proposed a TextRank-based AKE method that benefits from domain knowledge by using author-assigned keywords of scientific publications, and Gazendam et al. [30] proposed to use semantic relations between thesaurus terms for ranking candidate keywords without a reference corpus. Thesaurus relations have also been combined with machine learning techniques to improve the performance of AKE methods [31,32]. More recently, Sheoran et al. [33] leveraged domain-specific ontologies for aspect assignment of candidate keywords extracted from opinionated texts so that the selected candidates cover a maximum number of aspects.

Some AKE methods also make use of Wikipedia, a useful source of semantic information. Shi et al. [34] utilised Wikipedia to extract semantic features of candidate keywords. Their method constructs a semantic graph connecting candidate keywords to document topics based on the hierarchical relations extracted from Wikipedia, and semantic feature weights are assigned to candidate keywords with a link analysis algorithm. WikiRank is another AKE method leveraging Wikipedia [35]. It employs the TAGME annotator [36] to link meaningful word sequences in the input document to concepts in Wikipedia and constructs a semantic graph. Then, it transforms the KE task to an optimisation problem on the graph and tries to obtain the optimal keyword set that has the best coverage of the identified concepts. Finally, several embedding-based AKE methods utilise Wikipedia for pre-training and/or fine-tuning their underlying embedding methods [17,37].

Compared with existing AKE methods that have considered PoS tagging or semantic information more explicitly, our proposed approach is more universal and can be applied to *any* AKE method as a post-processor, which simply re-evaluates candidate keywords generated by an AKE method before the top n keywords are returned. Our approach is easily generalisable and can be used flexibly to eliminate candidate keywords that are unlikely to be keywords and prioritise those that are more likely to be keywords. Furthermore, our approach addresses the limitations of existing AKE approaches regarding semantic-awareness, mentioned in the previous subsections, without adding significant computational cost.

3. Analysis of Human-Annotated Keywords

Our proposed approach was motivated by some of our observations regarding how human labellers extracted "golden" (i.e., ground truth) keywords in 17 KE datasets. Such observations also helped us to determine some specific details, such as the parameters used in our proposed approach. In the following, we describe the 17 datasets we used and the key observations.

3.1. Datasets Inspected

Considering the subjectivity of the keyword extraction task, a standard approach has not been established to follow for constructing keyword extraction datasets [38]. This brings an extreme diversity to the datasets constructed so far, which makes comprehensive tests of keyword extraction algorithms harder. Therefore, to achieve a better understanding of human-annotated keywords, we aimed to collect a wide range of representative KE datasets used in the literature. With this respect, research papers corresponding to SOTA AKE methods and relevant surveys have been searched on multiple research databases, including Google Scholar and Scopus, with the keywords "automatic keyword extraction" and "automatic keyphrase extraction". Then, the collected papers were reviewed to identify datasets used by other researchers, and the publicly available datasets were downloaded. Multiple collections of AKE datasets have been found through different GitHub repositories (Examples include https://github.com/LIAAD/KeywordExtractor-Datasets, https://github.com/boudinfl/ake-datasets, and https://github.com/SDuari/Keyword-Extraction-Datasets, all accessed on 8 July 2025). In total, we were able to collect 17 datasets covering multiple contexts, including agriculture, computer science and health, and several types of documents, such as scientific papers, news, theses and abstracts. However, we excluded datasets containing short-text documents, such as tweets, since they tend to contain fewer candidate keywords, which could negatively impact the informativeness of our analysis. In addition, we only selected English datasets to limit the scope of our study to the English language due to the lack of a sufficient amount of non-English datasets and English being the only language shared by the authors of this paper. Further details regarding the datasets can be seen in Table 1.

Table 1. Basic information about the 17 datasets.

Dataset	Content	Context	Size	Avg. # (Keys)	Abs. Keys	Annotators [1]
KPCrowd [39]	News	Misc.	500	48.92	13.5%	Readers
citeulike180 [40]	Paper	Misc.	183	18.42	32.2%	Readers
DUC-2001 [13]	News	Misc.	308	8.1	3.7%	Readers
fao30 [41]	Paper	Agr.	30	33.23	41.7%	Experts
fao780 [41]	Paper	Agr.	779	8.97	36.1%	Experts
Inspec [26]	Abstract	CS	2000	14.62	37.7%	Experts
KDD [42]	Abstract	CS	755	5.07	53.2%	Authors
KPTimes (test) [43]	News	Misc.	20,000	5.0	54.7%	Editors
Krapivin2009 [44]	Paper	CS	2304	6.34	15.3%	Authors
Nguyen2007 [45]	Paper	CS	209	11.33	17.8%	Authors & Readers
PubMed [46]	Paper	Health	500	15.24	60.2%	Authors
Schutz2008 [47]	Paper	Health	1231	44.69	13.6%	Authors
SemEval2010 [48]	Paper	CS	243	16.47	11.3%	Authors & Readers
SemEval2017 [49]	Paragr	Misc.	493	18.19	0.0%	Experts & Readers
theses100 [2]	Thesis	Misc.	100	7.67	47.6%	Unknown
wiki20 [50]	Report	CS	20	36.50	51.2%	Readers
WWW [42]	Abstracts	CS	1330	5.80	55.0%	Authors

[1] Experts: Professional indexers assigned for annotation, Readers: People recruited for annotation regardless of their expertise, Authors: The authors of the document annotated. [2] https://github.com/LIAAD/KeywordExtractor-Datasets#theses100 (accessed on 8 July 2025).

3.2. Observations: PoS Tag Patterns

There has been a lot of research on linguistic properties of different multi-word expression types, such as collocations [51] and technical terms [52]. In addition, various PoS tag patterns have been proposed in the literature to identify noun phrases, which have been commonly considered a major indicator of keyword candidates [53]. However, these are unable to properly explain the linguistic properties of *keywords* used in AKE research because of the lack of linguistic standards for human-annotated keywords. Therefore, firstly, we reviewed the structure of human-annotated keywords in the 17 datasets, in terms of the used PoS tag patterns. For this purpose, we used the NLTK [54] library's PoS tagger and computed the distribution of different PoS tag patterns. As shown in Table 2, nine of the top ten PoS tag patterns correspond to either noun or gerund phrases. The only non-noun/gerund pattern in the top ten PoS tag patterns is a single adjective (JJ), with an average percentage of 6.85%. The top ten PoS tag patterns count 80% of all patterns. These observations imply that leveraging knowledge about how human labellers define keywords based on PoS tag patterns for a specific domain can potentially help improve the performance of any AKE methods for the corresponding domain.

3.3. Observations: n-Gram Size

AKE methods generally include a parameter for the maximum n-gram size, corresponding to the maximum number of words a keyword is allowed to contain. Although it is well-known that multi-word expressions (MWEs) are more likely to be of length two to three in English [55], it is less clear how human labellers of the 17 datasets were instructed to consider the n-gram size. Therefore, we analysed the golden keywords across the 17 datasets to see how human labellers decided on the n-gram sizes. On average, bigrams ($n = 2$) constitute 45.55% of the golden keywords in the 17 datasets, while this rate is 36.45% for unigrams ($n = 1$) and 12.73% for trigrams ($n = 3$). In addition, the percentages for keywords with $n \geq 4$ are considerably low—5.12% on average. More detailed statistics can be seen in Table 3. These results show that human labellers largely used two or three as the maximum n-gram size, covering 82.01% and 94.74% of the golden keywords across the different datasets, respectively. The results are aligned with those in the research literature

on MWEs. Based on such observations, we can see that AKE methods could benefit from focusing more on keywords with a shorter word length.

Table 2. Percentages of top 10 PoS tag patterns across 17 datasets. PoS tags: NN—noun (singular), NNS—noun (plural), JJ—adjective, VBG—verb gerund.

Dataset	NN	NN NN	JJ NN	NNS	JJ	JJ NNS	NN NNS	JJ NN NN	VBG	NN NN NN
KPCrowd	31.38	2.18	3.29	11.65	10.13	0.95	0.95	0.26	5.27	0.17
citeulike180	48.71	7.03	4.78	12.93	12.74	1.61	1.56	0.15	1.95	0.05
DUC-2001	19.13	15.90	15.28	10.49	1.80	8.73	10.16	3.65	0.28	1.52
fao30	32.60	14.68	7.92	15.84	5.06	6.62	9.35	0.00	0.78	0.26
fao780	29.56	14.11	9.11	15.18	3.78	6.02	10.88	0.06	1.21	0.04
Inspec	19.05	12.57	12.49	6.64	3.85	8.11	5.95	4.35	1.11	2.50
KDD	27.93	13.49	9.06	5.89	9.25	5.13	3.55	2.22	4.81	0.76
KPTimes	15.32	16.65	15.67	4.27	2.83	8.62	6.26	2.92	1.76	1.51
Krapivin2009	35.15	4.70	4.06	14.14	5.67	2.17	1.47	0.27	0.95	0.17
Nguyen2007	20.85	19.83	11.31	4.84	2.53	4.79	3.37	3.06	1.51	2.66
PubMed	30.88	9.23	3.87	15.43	12.01	3.51	5.50	0.77	0.56	2.03
Schutz2008	30.15	6.20	10.61	18.63	10.91	5.04	3.19	1.61	0.31	0.66
SemEval2010	19.45	21.74	21.54	0.08	3.20	0.17	0.06	6.40	0.42	3.15
SemEval2017	14.57	8.73	9.00	7.23	2.12	5.95	4.46	3.31	0.66	1.62
theses100	27.88	8.55	5.39	9.48	15.24	6.13	4.28	0.00	1.30	0.19
wiki20	41.91	18.65	11.06	1.49	6.60	0.50	1.82	2.81	2.81	0.99
WWW	32.33	13.44	8.98	5.41	8.74	3.88	3.88	1.63	2.86	1.05
Average (%)	*28.05*	*12.22*	*9.61*	*9.39*	*6.85*	*4.59*	*4.51*	*1.97*	*1.68*	*1.13*

Table 3. n-gram distributions of the 17 datasets. Bold values indicate the proportion of the most frequently observed n-gram length in the corresponding dataset.

Dataset	$n = 1$	$n = 2$	$n = 3$	$n \geq 4$	$n = 1, 2$	$1 \leq n \leq 3$
KPCrowd	**73.78**	18.47	4.90	2.83	92.25	97.15
citeulike180	**77.10**	19.98	2.79	0.09	97.08	99.87
DUC-2001	17.32	**61.29**	17.73	3.66	78.61	96.34
fao30	43.02	**52.74**	3.41	0.83	95.76	99.17
fao780	42.32	**53.72**	3.62	0.34	96.04	99.66
Inspec	16.44	**53.68**	23.05	6.84	70.12	93.17
KDD	25.48	**56.32**	13.97	4.24	81.80	95.77
KPTimes	**46.68**	34.39	12.55	6.38	81.07	93.62
Krapivin2009	18.95	**61.61**	15.74	3.70	80.56	96.30
Nguyen2007	27.53	**49.96**	15.42	6.97	77.49	92.91
PubMed	35.79	**43.74**	15.90	4.58	79.53	95.43
Schutz2008	**57.83**	30.22	8.15	1.67	88.05	96.20
SemEval2010	20.05	**52.97**	20.66	6.31	73.02	93.68
SemEval2017	25.23	**33.74**	17.19	23.84	58.97	76.16
theses100	31.63	**50.37**	11.09	6.90	82.00	93.09
wiki20	26.20	**53.52**	18.17	2.11	79.72	97.89
WWW	34.36	**47.71**	12.15	5.78	82.07	94.22
Average (%)	*36.45*	***45.55***	*12.73*	*5.12*	*82.01*	*94.74*

3.4. Observations: Semantic Information

Finally, we analysed the human-annotated keywords to see if human labellers explicitly or implicitly relied on semantic information to select keywords. We first calculated the percentage of golden keywords that are covered by Wikipedia across all the datasets. This quantitative analysis indicated that, on average, 64.39% of the golden keywords are Wikipedia named entities, i.e., titles of Wikipedia articles. This interesting (previously unreported) finding justifies that Wikipedia can be a very useful knowledge base for AKE algorithms, as it covers so many golden keywords chosen by human labellers for all the 17

datasets we chose. Although unexpected, this finding can be explained by the diversity and richness of the content of Wikipedia. More detailed results of the analysis can be seen in Table 4.

Table 4. The percentages of golden keywords covered by Wikipedia.

Dataset	%	Dataset	%
KPCrowd	71.77	Nguyen2007	52.19
citeulike180	83.78	PubMed	81.28
DUC-2001	51.05	Schutz2008	67.43
fao30	80.97	SemEval2010	41.27
fao780	79.00	SemEval2017	31.02
Inspec	39.08	theses100	68.82
KDD	62.92	wiki20	89.01
KPTimes	79.09	WWW	63.83
Krapivin2009	52.12		

In addition to Wikipedia named entities, we also manually inspected many golden keywords and observed that many collected datasets contain domain-specific golden keywords. This observation indicates that considering domain-specific terms can potentially help improve the performance of AKE methods, too.

4. Methodology

4.1. Problem Definition and Our Proposed Approach

Suppose $W^C(D) = \{w_i^C\}_{i=1}^m$ denotes m candidate keywords generated from a document D by an AKE method. In addition, let $W^S(D) \subset W^C(D)$ denotes $n \leq m$ keywords produced by the AKE method. Finally, let $W(D) = \{w_i\}_{i=1}^t$ be the set of ground truth keywords an ideal AKE method should extract from D. Given the above notations, our goal is to find post-processing methods that can minimise $|W(D) - W^S(D)|$ (false negatives) and $|W^S(D) - W(D)|$ (false positives). Among the two types of errors, reducing false negatives is more important than reducing false positives, but given the fact that n cannot be too large to make the results manageable, balancing both types of errors is still very important. Typically, AKE methods select keywords by assigning a numerical score s_i to each candidate keyword w_i, and then return the top n keywords with the highest (Although some AKE methods, e.g., YAKE!, use smaller scores for better keywords, here, for the sake of simplicity, we assume that a higher score means a more preferred keyword.) scores. Our proposed approach can work with any AKE method with such a scoring system, and it aims to re-adjust such scores so that true positive keywords' scores will more likely increase and true negative keywords' scores will more likely decrease.

Informed by the findings presented in Chapter 3, our proposed approach is based on three general post-processing steps that can be applied to any baseline AKE methods as shown in Figure 1: (1) removing candidate keywords with an unlikely PoS tag pattern by zeroing its score ($s_i = 0$), (2) using one or more context-aware (i.e., domain-specific) thesauri to prioritise important candidate keywords for the target domain ($s_i = c_i s_i$, where c_i is an amplifying factor larger than 1), and (3) prioritising candidate keywords that are Wikipedia named entities ($s_i = w_i s_i$, where w_i is another amplifying factor larger than 1). Note that the amplifying factor c_i and w_i can be a static value for all prioritised keywords (so independent of i) or a keyword-dependent factor, depending on an importance score of each candidate keyword in the thesauri and Wikipedia, e.g., c_i can be proportional to the word frequency in the thesauri and w_i can be proportional to the size of the Wikipedia entry or the number of references to the entry.

In the following subsections, we explain the three steps in more detail.

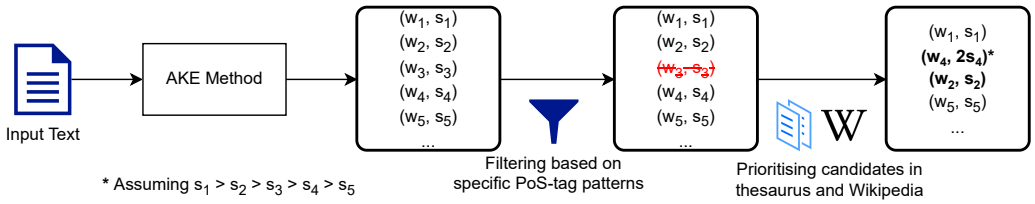


Figure 1. The overview of the proposed post-processing approach.

4.2. Filtering Specific PoS Tag Patterns

As mentioned in Section 2.3, PoS tagging has been extensively used in AKE methods to consider morpho-syntactic features. Motivated by the observations in Section 3.2, we attempted to leverage a PoS tagger to filter out candidate keywords labelled with unlikely PoS tag patterns. More precisely, candidate keywords that do not conform with any of the following PoS tag patterns were discarded: (i) *simple nouns and noun phrases*—one or more nouns/gerunds (optionally with one or more adjectives appearing before the first noun); (ii) two or more *simple nouns and/or noun phrases* connected by one or more prepositions or conjunctions (Examples include “quality of service" and “buyer and seller" from the SemEval2010 dataset. Although none of the possible PoS tag patterns conforming to this criterion are among the most common patterns presented in Table 2 individually, they collectively constitute 1.3% of all patterns across the 17 datasets.); and (iii) a single adjective.

In the PoS tag patterns mentioned above, *nouns* and *adjectives* mean any PoS tags that can provide the corresponding functionality in a sentence. Therefore, nouns also include gerunds, and adjectives also include past participle verbs. Considering the most common PoS tag patterns mentioned in Section 3.2, our proposed PoS tag patterns correspond to over 90% of the patterns observed across all the 17 datasets. We used NLTK to extract PoS tags for each term in the input documents. For pattern matching, we took advantage of regular expressions. Since regular expressions return the longest possible matches, we extracted the shorter matches from the longest ones separately.

Note that the proposed PoS tag patterns can be further changed to reflect any domain-specific needs, e.g., we observed that gerunds are quite uncommon in the health domain, so they can be removed if preferred.

4.3. Context-Aware Thesauri

Context means any kind of domain, topic or field that has its own set of terms semantically specific to itself. While the set of terms specific to a context can be covered in a more structured vocabulary, such as a thesaurus or an ontology, a simple word list can often be sufficient for the purpose of AKE. As reported in Section 3.4, many keywords are related to the context of the input text, and contextual consideration can be quite useful for AKE. Therefore, we propose to make use of external resources to inform AKE methods more about semantically useful keywords for the relevant domain. More specifically, we proposed to integrate one or more domain-specific thesauri, which contain terms specific to a target context, and to prioritise candidate keywords included in such thesauri. At the implementation level, we introduce a weight for each candidate keyword and increase the weight of any candidate keyword appearing in one of the thesauri. In our experiments, we doubled the weights of candidate keywords in a thesaurus. However, the actual weight increase can be a parameter that can be empirically determined based on some training data or qualitative evidence observed. To determine if a candidate keyword exists in a given thesaurus, we applied exact matching with lemmatisation. Although using stemming with exact matching is a more common practice in AKE [3], we preferred to use lemmatisation due to its context-awareness. In our experiments, we focused on thesauri with a single con-

text, but using multiple contexts in a single thesaurus is of course also possible. Regarding integrating relevant thesauri, we considered two different approaches explained below.

Manual Context Consideration:

This approach is more useful when documents processed by an AKE method are known to belong to a specific context. It utilises one or more thesauri containing a list of terms relevant to the context, which are given a higher weight for prioritisation by the AKE method. In our experiments, we assigned a single domain-specific thesaurus to each of the datasets to represent the relevant context. Note that it is possible that multiple contexts and multiple thesauri are used in some applications of AKE.

Automatic Context Identification:

Considering the wide range of applications in which AKE methods can be utilised, manually providing a thesaurus for each input document may not be very usable. Therefore, we also studied how to identify the context of an input document automatically, which can allow assigning a different context and a corresponding thesaurus automatically. This can be achieved by building a machine learning-based classifier, which produces a class label representing the context or a context-specific thesaurus of a given document or its abstract.

Once the classifier predicts the context of an input abstract, we identify a thesaurus corresponding to the context, as defined in a context-to-thesaurus look-up table, to inform the AKE method. Unlike the manual approach, automatic identification allows us to use a different thesaurus for each document in the dataset; therefore, it can be applied to many real-world scenarios where the documents processed can belong to multiple contexts.

4.4. Wikipedia Named Entities

Based on the Wikipedia-related observations reported in Section 3.4, we propose to use Wikipedia as a *context-independent* thesaurus to improve the performance of any AKE methods working in any context(s). Similar to how a thesaurus can be used, we prioritise the candidate keywords covered by Wikipedia as an entry by increasing their weight. Then, we apply exact matching with lemmatisation to identify if a candidate keyword is a Wikipedia named entity. Since Wikipedia also contains a vast amount of entries with too general semantic meanings, e.g., unigrams such as 'father', 'school', and 'table' that are normally already well covered by most AKE methods, we utilised the NLTK's *words* corpus (i.e., a wordlist including common English dictionary words) to identify such unigrams and remove them from the Wikipedia entities that will be prioritised in our post-processing step. For the Wikipedia named entities, we used the 2021-10-01 version of the English Wikipedia dump (https://archive.org/download/enwiki-20211001, accessed on 8 July 2025), containing only page titles. We first cleaned the dump data by removing the disambiguation tags (https://en.wikipedia.org/wiki/Wikipedia:Disambiguation#Naming_the_disambiguation_page, accessed on 8 July 2025) added next to the title by Wikipedia. Then, we normalised the data with lemmatisation and lower-casing by following the common practice.

5. Experiments and Results

5.1. Evaluation Metrics

As the evaluation metrics, we used precision, recall and F1 score at the top ten keywords, which have been commonly used in AKE evaluation [3]. Furthermore, we adopted micro-averaging and exact matching with stemming when calculating the scores.

5.2. Selecting Baseline Methods

To show the effectiveness and generalisability of the proposed methods, we first attempted to identify some representative AKE algorithms with different key characteristics for our experiments. We reviewed existing AKE algorithms in terms of multiple aspects, e.g., recency, ease of reconfiguration, and whether they already use one or more of our proposed methods by any means, as shown in Table 5. These methods are considered more representative because they have open-source implementations, are applicable to any document type, were validated on a number of datasets, and do not require training (i.e., unsupervised so that it is easier to use and less likely to have generalisation problems) (Unsupervised AKE methods have become more popular for this reason. Most implementations of supervised methods are also harder to reconfigure.). Among these methods, we selected two statistical methods, i.e., KP-Miner and YAKE!, two graph-based methods, i.e., RaKUn and LexRank, and an embedding-based method, i.e., SIFRank+, as baseline methods for our experiments. Since SIFRank+ is very computationally costly, we used only seven of the datasets, containing shorter documents (i.e., KPCrowd, DUC-2001, Inspec, KDD, KPTimes, SemEval2017 and WWW) for its evaluation when our methods were applied.

For the implementations of the selected AKE methods, we utilised the PKE [56] library for KP-Miner and the original implementations of the other four. We used the default parameters for all the methods, except the maximum n-gram size parameter. Considering the n-gram size across the datasets being mostly limited up to 3, as mentioned in Section 3.3, we set the maximum n-gram size to be 3.

Table 5. An overview of some existing open-source unsupervised AKE methods, showing a number of key characteristics.

Method	Easy to Reconfigure	PoS tagging	Thesaurus	Wikipedia
Statistical Methods				
KP-Miner [6]	✓	–	–	–
YAKE! [5]	✓	–	–	–
LexSpec [8]	✓	✓	–	–
Graph-based Methods				
TextRank [12]	✓	✓	–	–
SingleRank [13]	✓	✓	–	–
RAKE [14]	✓	–	–	–
RaKUn [7]	✓	–	–	–
LexRank [8]	✓	✓	–	–
TFIDFRank [8]	✓	✓	–	–
Embeddings-based Methods				
EmbedRank [17]	✓	✓	–	✓
SIFRank [9]	✓	✓	–	✓
SIFRank+ [9]	✓	✓	–	✓
MDERank [18]	–	✓	–	✓

5.3. PoS Tag Patterns

As the first step, we applied our PoS tagging-based post-processing approach to the selected AKE methods and evaluated on all the datasets. Results show that the proposed approach improved all the methods except SIFRank+ on average, in terms of precision, recall, and F1 score. While KP-Miner achieved better performance for 14 of the 17 datasets with an average of 6.08% in F1 score, RaKUn was improved by a cross-dataset average of 4.46% for 14 of the 17 datasets. We observed the most change in the performance of

YAKE!—it was improved in 16 of the 17 datasets by a cross-dataset average of 18.05%. We believe this is because YAKE! does not benefit from linguistic features as a more language-independent (multilingual) approach. Finally, we observed a limited improvement in the scores of LexRank (0.84% on average) for 12 of the 17 datasets, and a slight decrease in the performance of SIFRank+, which is likely due to the fact that these two methods already use PoS tagging-based filtering. The obtained scores for YAKE! and SIFRank+ are shown in Tables 6 and 7, respectively, as examples. These results provide new evidence for the effectiveness of PoS tagging in AKE algorithms and imply that there is still room to improve the use of PoS tagging in many AKE methods.

Table 6. Comparison of the precision, recall, and F1 score of the original YAKE! and the one utilising PoS tagging, at 10 extracted keywords. Bold values indicate the best scores obtained for each dataset.

Dataset	YAKE!			YAKE!+PoS		
	P%	R%	F1%	P%	R%	F1%
KPCrowd	24.20	4.92	8.17	**33.98**	**6.90**	**11.47**
citeulike180	23.11	13.27	16.86	**25.68**	**14.74**	**18.73**
DUC-2001	12.01	14.87	13.29	**17.44**	**21.58**	**19.29**
fao30	22.00	6.83	10.42	**25.33**	**7.86**	**12.00**
fao780	11.93	14.95	13.27	**13.18**	**16.52**	**14.67**
Inspec	19.82	14.05	16.44	**24.57**	**17.41**	**20.38**
KDD	**6.01**	**14.68**	**8.53**	5.83	14.23	8.27
KPTimes	7.97	15.83	10.61	**11.37**	**22.58**	**15.12**
Krapivin2009	9.54	17.88	12.44	**9.93**	**18.61**	**12.95**
Nguyen2007	19.00	15.82	17.26	**19.19**	**15.98**	**17.43**
PubMed	7.28	5.11	6.01	**8.66**	**6.08**	**7.15**
Schutz2008	37.29	8.06	13.26	**47.63**	**10.30**	**16.93**
SemEval2010	20.37	13.08	15.93	**20.82**	**13.37**	**16.28**
SemEval2017	20.61	11.91	15.10	**29.41**	**17.00**	**21.55**
theses100	9.40	14.09	11.28	**10.50**	**15.74**	**12.60**
wiki20	19.50	5.49	8.57	**22.00**	**6.20**	**9.67**
WWW	6.49	13.47	8.76	**6.58**	**13.66**	**8.88**
Avg. Score (%)	16.27	12.02	12.13	**19.54**	**14.04**	**14.32**
Improvement (%)				*20.10*	*16.81*	*18.05*

Table 7. Comparison of the precision, recall, and F1 score of the original SIFRank+ and the one utilising PoS tagging, at 10 extracted keywords. Bold values indicate the best scores obtained for each dataset.

Dataset	SIFRank+			SIFRank+ + PoS		
	P%	R%	F1%	P%	R%	F1%
KPCrowd	26.08	5.30	8.81	**26.20**	**5.32**	**8.85**
DUC-2001	**28.34**	**35.09**	**31.36**	27.86	34.49	30.82
Inspec	**35.68**	**25.29**	**29.60**	35.10	24.88	29.12
KDD	**5.68**	**13.87**	**8.06**	4.42	10.80	6.28
KPTimes	**7.92**	**15.74**	**10.54**	7.74	15.37	10.30
SemEval2017	**41.66**	**24.08**	**30.52**	40.16	23.21	29.42
WWW	**6.59**	**13.69**	**8.90**	5.26	10.93	7.10
Avg. Score (%)	**21.71**	**19.01**	**18.26**	20.96	17.86	17.41
Improvement (%)				*−3.45*	*−6.05*	*−4.65*

Finally, we studied how tailoring the selected PoS tag patterns according to domain-specific needs may affect the performance of AKE methods. To this end, we considered the

example given in Section 4.2, i.e., the observation that gerunds are rarely seen as a keyword in the health domain. We selected the health datasets (i.e., PubMed and Schutz2008) from our collection and applied the tailored PoS tag-based filtering that disregards gerunds. For this experiment, we used YAKE! since it is more sensitive to linguistic-based improvements as a language-independent algorithm. As shown in Table 8, the tailored filtering approach provided some small improvements to our original filtering proposal in terms of precision, recall, and F1 score. The limited improvement is likely due to the small percentage of gerunds as candidate keywords.

Table 8. Comparison of the precision, recall, and F1 score of YAKE! when the original (PoS) and the tailored (PoS*) filtering approaches are used, at 10 extracted keywords. Bold values indicate the best scores obtained for each dataset.

Dataset	YAKE! + PoS			YAKE! + PoS*		
	P%	R%	F1%	P%	R%	F1%
PubMed	8.66	6.08	7.15	**8.70**	**6.11**	**7.18**
Schutz2008	47.63	10.30	16.93	**47.80**	**10.34**	**17.00**
Avg. Score (%)	28.15	8.19	12.04	**28.25**	**8.23**	**12.09**
Improvement (%)				*0.36*	*0.49*	*0.42*

5.4. Context-Aware Thesauri

For this step, we selected 10 datasets mentioned in Section 3.1 that have a particular context. The included contexts (and datasets) are agriculture (fao30 and fao780), health (PubMed) and computer science (Inspec, Krapivin2009, Nguyen2007, SemEval2010, KDD, Wiki20 and WWW). In addition, we constructed another context-specific dataset, KPTimes-Econ, by extracting economy-related news from the KPTimes dataset, which includes 3258 news articles. For extracting economy-related news articles, we have looked for the records involving the term "economy" in the *keyword* and/or *categories* field(s). Based on the 11 datasets, we collected a thesaurus (or something similar, e.g., dictionary, ontology, or wordlist) for each context. More specifically, we used the following thesauri: (i) AGROVOC 2021-07 [57]—a multilingual controlled vocabulary constructed by the Food and Agriculture Organization of the United Nations (FAO), with 844,000 agriculture-related terms including 50,163 English ones; (ii) Medical Subject Headings (MeSH) 2021 [58]—a thesaurus covering biomedical and health-related terms produced by the National Library of Medicine (NLM), with over 1.4 million terms in English; (iii) Computer Science Ontology (CSO) v3.3 [59]—a large-scale computer science ontology automatically produced by Klink-2 [60] algorithm from 16 million computer science publications, with 14,000 terms; and (iv) STW v9.10 [61]—a bilingual thesaurus (in English and German) for economics produced by the Leibniz Information Center for Economics (ZBW), with over 20,000 terms including 6217 English ones.

For the initial step aiming to experiment with manual integration, we fed each of the baseline methods with each of the datasets and their corresponding thesaurus depending on the context. As in the previous experiment, SIFRank+ was evaluated on only the datasets with shorter documents, i.e., Inspec, KDD, WWW, and KP-Times-Econ in this case. The experiments showed that the manual integration of context-aware thesaurus improved all five AKE methods in terms of precision, recall, and F1 score significantly for all the datasets. The improvement in F1 score was observed to be 29.03%, 23.88%, 12.85%, 13.19%, and 7.09% for RaKUn, LexRank, YAKE!, KP-Miner, and SIFRank+, respectively. Tables 9 and 10 show more detailed results of the experiment for LexRank and SIFRank+, respectively. The results of this experiment produced solid evidence of the effectiveness of using context-aware thesaurus to improve the performance of AKE methods.

Table 9. Comparison of precision, recall, and F1 score of the original LexRank and its enhanced versions with manual (M) and automatic (A) thesaurus integration, at 10 extracted keywords. Bold values indicate the best scores obtained for each dataset.

Dataset	Context	LexRank			LexRank + T (M)			LexRank + T (A)		
		P%	R%	F1%	P%	R%	F1%	P%	R%	F1%
fao30	Agr.	20.33	6.31	9.63	**30.33**	**9.41**	**14.36**	—	—	—
fao780	Agr.	8.55	10.72	9.51	**13.04**	**16.35**	**14.51**	—	—	—
Inspec	CS	30.49	21.61	25.29	**31.10**	**22.04**	**25.79**	30.97	21.95	25.69
KDD	CS	6.07	14.81	8.61	6.23	15.20	8.83	**6.25**	**15.26**	**8.87**
Krapivin2009	CS	7.01	13.14	9.15	**8.79**	**16.48**	**11.47**	8.74	16.37	11.39
Nguyen2007	CS	13.25	11.04	12.04	**15.69**	**13.07**	**14.26**	15.45	12.87	14.04
SemEval2010	CS	13.13	8.43	10.27	**15.10**	**9.70**	**11.81**	**15.10**	**9.70**	**11.81**
wiki20	CS	14.00	3.94	6.15	**23.00**	**6.48**	**10.11**	**23.00**	**6.48**	**10.11**
WWW	CS	6.66	13.83	8.99	**6.95**	**14.43**	**9.38**	6.93	14.40	9.36
PubMed	Health	4.22	2.96	3.48	**8.98**	**6.31**	**7.41**	8.92	6.26	7.36
Schutz2008	Health	28.32	6.12	10.07	**34.35**	**7.43**	**12.21**	34.00	7.35	12.09
KPTimes-Econ	Econ.	3.27	7.03	4.46	**4.09**	**8.80**	**5.59**	4.09	8.79	5.58
Avg. Score (%)		12.94	9.99	9.80	**16.47**	**12.14**	**12.14**	15.35	11.94	11.63
Improvement (%)					*27.28*	*21.52*	*23.88*	*21.44*	*16.03*	*18.07*

Table 10. Comparison of precision, recall, and F1 score of the original SIFRank+ and its enhanced versions with manual (M) and automatic (A) thesaurus integration, at 10 extracted keywords. Bold values indicate the best scores obtained for each dataset.

Dataset	Context	SIFRank+			SIFRank+ + T (M)			SIFRank+ + T (A)		
		P%	R%	F1%	P%	R%	F1%	P%	R%	F1%
Inspec	CS	35.68	25.29	29.60	**36.62**	**25.95**	**30.37**	36.03	25.53	29.88
KDD	CS	5.68	13.87	8.06	**5.97**	**14.58**	**8.48**	5.95	14.52	8.44
WWW	CS	6.59	13.69	8.90	**7.32**	**15.19**	**9.88**	7.27	15.10	9.81
KPTimes-Econ	Econ.	3.49	7.50	4.76	**4.56**	**9.81**	**6.23**	**4.56**	**9.81**	**6.23**
Avg. Score (%)		12.86	15.09	12.83	**13.62**	**16.38**	**13.74**	13.45	16.24	13.59
Improvement (%)					*5.91*	*8.55*	*7.09*	*4.59*	*7.62*	*5.92*

For the next step, we experimented with the automated thesauri integration process. In our experiments, especially for datasets covering mainly scientific papers, we built a classifier for classifying a given article's title and abstract into the main discipline the article belongs to. The classifier was trained on samples extracted from the arXiv.org dataset (https://www.kaggle.com/Cornell-University/arxiv) containing metadata of over 1.7M preprints in multiple disciplines. Before the training process, we filtered the arXiv.org dataset by the main discipline reflected by its *categories* field so as to include the following three disciplines: (1) cs (Computer Science, e.g., cs.AI), (2) bio (Biology, e.g., q-bio), and (3) fin (Finance, e.g., q-fin.CP) and econ (Economics, e.g., econ.EM). After this filtering process, we obtained a dataset of 583,796 samples (551,443 computer science, 20,110 biology, and 12,243 finance/economics samples). Since the resulting dataset is highly imbalanced, we applied random downsampling to equate the number of samples from each discipline to the size of the smallest class, 12,243, which made the final size of our training set 36,729. In our classifier, we utilised the TF-IDF vectoriser for feature extraction. We chose to use the calibrated linear support vector classifier (SVC) with the default parameters and the one-vs-rest setting, rather than a multi-class classification method or more advanced feature extraction methods such as BERT, to show that even a lightweight classifier is sufficient for the task of automatic context detection. The classifier was evaluated with a

stratified 5-fold cross-validation. The testing accuracies (i.e., the fraction of the number of correct predictions with respect to the total number of predictions) of computer science, biology and finance/economics models were 93.2%, 94.9%, and 97.0%, respectively. The classifier can also be extended to support multiple contexts for a single article, although in our experiments, we considered the case of a single context per article for the sake of simplicity and clarity. We used the Scikit-learn library [62] to implement all of the mentioned components.

Since the training set of the classifier does not cover agriculture preprints, and we were unable to find a proper agriculture dataset for training, we excluded the agriculture context and the corresponding datasets, fao30 and fao780, for this part of the experiments. The results of the experiments performed with our classifier indicated that the automatic thesaurus integration approach achieved as good as the manual integration approach with a negligible performance decrease. More precisely, the F1 score was improved by an average of 23.23%, 18.07%, 9.60%, 11.27%, and 5.92% for RaKUn, LexRank, YAKE!, KP-Miner, and SIFRank+, respectively, compared to the baseline scores. Tables 9 and 10 show more detailed results of the experiment for LexRank and SIFRank+, respectively. The obtained results imply that automatic integration can be generalised to cover more contexts and thesauri, which can be quite useful in real-world AKE applications.

5.5. Wikipedia Named Entities

For this part of the experiments, we used the entire set of datasets as we did in Section 3.2. The results of the experiment indicated that leveraging Wikipedia named entities improved the performance of KP-Miner and RaKUn for 16 of the datasets, and the performance of YAKE! and LexRank for all the datasets, in terms of all the evaluation metrics. Furthermore, the average improvement rates of the F1 score were observed as 18.83%, 11.11%, 10.96%, and 10.11% for RaKUn, LexRank, YAKE!, and KP-Miner, respectively. However, we observed a slight decrease in the average F1 score of SIFRank+, although it improved for most (5 out of 7) of the datasets, which may be explained by its underlying sentence embedding approach, SIF [63], which already leverages Wikipedia for pre-training and fine-tuning. Tables 11 and 12 show more detailed results for RaKUn and SIFRank+ as examples.

5.6. Combining Post-Processing Steps

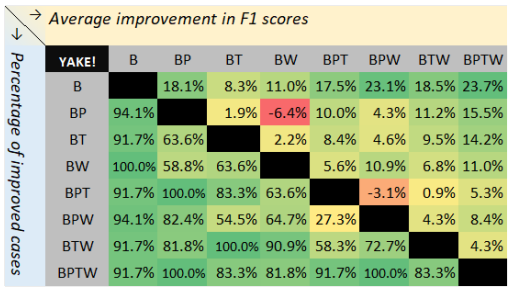
In the final part of our experiments, we tried combining multiple post-processing steps to improve the performance further. With this respect, we tried to apply all the combinations of the three proposed enhancements. The generated heatmaps from the F1@10 scores and the percentages of improved cases with different combinations for each baseline method can be seen in Figure 2. The results show that the best F1 scores for YAKE!, RaKUn, and KP-Miner were obtained when all the proposed post-processing steps were applied. For LexRank and SIFRank+, however, the best combination was integrating context-aware thesaurus and Wikipedia since they already benefited from PoS tagging-based filtering. In addition, the applied post-processing steps improved the baselines significantly –the improvement rate reached up to 23.7% for YAKE!, 21.3% for KP-Miner, 53.8% for RaKUn, 20.1% for LexRank, and 10.2% for SIFRank+. Finally, the improvements were consistent—at least one combination of the post-processing steps was observed for each method, resulting in higher performance across all the datasets. The results showed that even for more modern AKE methods there is still room for improvement using simple post-processing steps like those proposed in this paper.

Table 11. Comparison of precision, recall, and F1 score of the original RaKUn and its enhanced versions with Wikipedia, at 10 extracted keywords. Bold values indicate the best scores obtained for each dataset.

Dataset	RaKUn			RaKUn+Wiki		
	P%	R%	F1%	P%	R%	F1%
KPCrowd	42.52	8.64	14.36	**42.64**	**8.66**	**14.40**
citeulike180	16.56	9.50	12.08	**17.92**	**10.29**	**13.07**
DUC-2001	5.68	7.03	6.29	**6.17**	**7.64**	**6.82**
fao30	15.00	4.65	7.10	**18.67**	**5.79**	**8.84**
fao780	6.50	8.14	7.23	**7.64**	**9.57**	**8.50**
Inspec	6.54	4.64	5.43	**6.74**	**4.77**	**5.59**
KDD	**3.66**	**8.92**	**5.19**	3.63	8.86	5.15
KPTimes	8.07	16.03	10.74	**8.15**	**16.18**	**10.84**
Krapivin2009	2.77	5.20	3.62	**4.94**	**9.26**	**6.44**
Nguyen2007	6.79	5.66	6.17	**9.67**	**8.05**	**8.78**
PubMed	4.30	3.02	3.55	**6.58**	**4.62**	**5.43**
Schutz2008	33.14	7.16	11.78	**40.09**	**8.67**	**14.25**
SemEval2010	6.75	4.33	5.28	**10.04**	**6.45**	**7.85**
SemEval2017	11.42	6.60	8.37	**11.74**	**6.79**	**8.60**
theses100	3.90	5.85	4.68	**4.80**	**7.20**	**5.76**
wiki20	9.50	2.68	4.18	**19.50**	**5.49**	**8.57**
WWW	4.32	8.98	5.84	**4.39**	**9.12**	**5.93**
Avg. Score (%)	11.02	6.88	7.17	**13.14**	**8.08**	**8.52**
Improvement (%)				*19.24*	*17.44*	*18.83*

Table 12. Comparison of the precision, recall, and F1 score of the original SIFRank+ and the one utilising Wikipedia named entities, at 10 extracted keywords. Bold values indicate the best scores obtained for each dataset.

Dataset	SIFRank+			SIFRank+ + Wiki		
	P%	R%	F1%	P%	R%	F1%
KPCrowd	26.08	5.30	8.81	**27.46**	**5.58**	**9.27**
DUC-2001	**28.34**	**35.09**	**31.36**	22.82	28.26	25.25
Inspec	35.68	25.29	29.60	**36.60**	**25.94**	**30.36**
KDD	5.68	13.87	8.06	**6.11**	**14.90**	**8.66**
KPTimes	7.92	15.74	10.54	**9.22**	**18.31**	**12.26**
SemEval2017	**41.66**	**24.08**	**30.52**	41.34	23.89	30.28
WWW	6.59	13.69	8.90	**7.50**	**15.57**	**10.12**
Avg. Score (%)	**21.71**	**19.01**	**18.26**	21.58	18.92	18.03
Improvement (%)				*−0.60*	*−0.47*	*−1.26*



(**a**) YAKE!

Figure 2. *Cont.*

Average improvement in F1 scores / Percentage of improved cases

KP-Miner	B	BP	BT	BW	BPT	BPW	BTW	BPTW
B		6.1%	13.2%	10.1%	16.5%	13.2%	20.9%	21.3%
BP	88.2%		7.3%	3.8%	10.8%	6.7%	14.7%	15.1%
BT	100.0%	63.6%		-2.8%	3.0%	-0.4%	6.8%	7.2%
BW	94.1%	76.5%	45.5%		6.1%	2.8%	9.9%	10.3%
BPT	100.0%	100.0%	66.7%	72.7%		-3.6%	3.8%	4.1%
BPW	100.0%	88.2%	54.5%	88.2%	36.4%		7.3%	7.7%
BTW	100.0%	90.9%	100.0%	100.0%	66.7%	72.7%		0.3%
BPTW	100.0%	100.0%	83.3%	81.8%	91.7%	100.0%	66.7%	

(**b**) KP-Miner

Average improvement in F1 scores / Percentage of improved cases

RaKUn	B	BP	BT	BW	BPT	BPW	BTW	BPTW
B		4.5%	29.0%	18.8%	33.3%	22.3%	49.4%	53.8%
BP	82.4%		22.5%	13.8%	27.1%	17.1%	43.1%	46.4%
BT	100.0%	90.9%		0.4%	3.3%	3.8%	15.8%	19.2%
BW	88.2%	82.4%	45.5%		3.4%	2.9%	16.4%	19.1%
BPT	83.3%	83.3%	75.0%	54.5%		0.0%	12.1%	15.4%
BPW	88.2%	94.1%	54.5%	76.5%	54.5%		12.6%	15.2%
BTW	100.0%	100.0%	91.7%	90.9%	75.0%	72.7%		3.0%
BPTW	83.3%	100.0%	83.3%	81.8%	100.0%	90.9%	83.3%	

(**c**) RaKUn

Average improvement in F1 scores / Percentage of improved cases

LexRank	B	BP	BT	BW	BPT	BPW	BTW	BPTW
B		0.8%	11.0%	11.1%	9.3%	11.1%	20.1%	18.9%
BP	70.6%		24.4%	10.2%	22.1%	10.2%	33.8%	32.2%
BT	100.0%	100.0%		-8.1%	-1.5%	-8.6%	8.2%	7.2%
BW	100.0%	100.0%	25.0%		6.8%	0.1%	17.0%	15.6%
BPT	83.3%	100.0%	41.7%	72.7%		-6.9%	9.9%	8.8%
BPW	82.4%	100.0%	25.0%	82.4%	27.3%		17.7%	16.3%
BTW	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%		-1.0%
BPTW	83.3%	100.0%	83.3%	81.8%	100.0%	100.0%	50.0%	

(**d**) LexRank

Average improvement in F1 scores / Percentage of improved cases

SIFRank+	B	BP	BT	BW	BPT	BPW	BTW	BPTW
B		-4.7%	7.1%	-1.3%	-2.3%	-7.0%	10.2%	1.3%
BP	14.3%		14.6%	3.6%	4.2%	-2.5%	16.4%	5.6%
BT	100.0%	100.0%		0.9%	-8.7%	-8.6%	2.9%	-5.5%
BW	71.4%	85.7%	66.7%		-9.9%	-5.8%	0.9%	-8.7%
BPT	50.0%	100.0%	0.0%	0.0%		0.1%	12.8%	3.6%
BPW	42.9%	71.4%	0.0%	14.3%	66.7%		11.1%	0.8%
BTW	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%		-8.1%
BPTW	50.0%	100.0%	25.0%	0.0%	100.0%	100.0%	25.0%	

(**e**) SIFRank+

Figure 2. Average improvements in F1 scores across all the datasets (upper side), and percentages of the improved cases across all the datasets (bottom side), for different AKE methods. (B: Baseline, P: PoS tagging, T: Thesaurus integration, W: Wikipedia integration).

6. Further Discussions

The proposed post-processing steps in this study were applied to five representative SOTA AKE methods, showing their universality to improve the performance of many different AKE methods. The universality of the post-processing steps is rooted in the fact that they rely on access to the list of candidate keywords and their scores, which are the standard output for most (if not all) AKE methods. The performance improvements can be explained by two main reasons: (i) utilising PoS tagging avoids AKE methods, especially those less benefiting from linguistic features, to generate keywords that are less likely to be meaningful keywords, such as conjunctions, determiners, and adverbs; and (ii)

thesauri and Wikipedia-based enhancements allow prioritisation of more domain-specific and context-specific keywords to be returned by AKE methods.

Although PoS tagging can be easily integrated into AKE methods to implement a filtering mechanism, it should be separately considered for each dataset since AKE datasets lack linguistic standards for golden keywords. This can significantly increase the accuracy of the AKE methods, benefiting from PoS tagging. Thesaurus and Wikipedia integration can also be applied to AKE methods without much effort. Considering that a text document can cover multiple contexts, the results we reported can be further improved by integrating multiple contexts. This can be achieved by utilising a multi-label classifier. Since one-vs-rest classifiers can be used for multi-label classification, our classifier can be refined to cover multiple contexts. In addition, more advanced models, such as BERT, can be utilised to develop a more accurate classifier. It is also worth noting that two of the proposed post-processing steps in this study were selected as representative examples of semantic elements. Other semantic elements can also be used to further improve the performance of AKE methods.

Although our experiments on the proposed post-processing steps are based on English NLP tools and datasets, they can also be applied to multilingual AKE methods, e.g., YAKE!, for any language. The language of input documents can be identified automatically with a language identifier, which can achieve high accuracy for many languages [64]. Then, the corresponding PoS tagger and Wikipedia data can be utilised, although the set of acceptable PoS tag patterns will need updating according to the identified language. Nevertheless, utilising a context-aware thesaurus could be tricky for some languages, especially small ones, as there might be no thesaurus relevant to the context of the document in the identified language.

This study has a number of limitations that can be addressed in future work. Firstly, the selected baseline AKE methods are just examples of SOTA methods, so they may not be sufficiently representative. As our focus was improving AKE methods in general, we did not aim to achieve the best scores among the studies on AKE. As a result, this study is limited to open-source, unsupervised, and general-purpose AKE methods. In addition, this study leveraged multiple elements of the English language and used English datasets for evaluation. Therefore, it disregarded non-English settings, which are needed especially for multilingual AKE methods, such as YAKE!. The proposed mechanisms have been applied separately throughout the experiments. Therefore, the results could be improved further if different mechanisms benefit from each other (e.g., applying PoS tag-based filtering to the Wikipedia integration mechanism to disregard the Wikipedia named entities that cannot be keywords). Finally, a better matching strategy considering word ambiguities can be developed for checking if a candidate keyword appears in a thesaurus or Wikipedia, with the help of techniques such as word sense disambiguation.

Furthermore, while the proposed post-processing approaches are designed to be applicable across a wide range of AKE methods, their effectiveness inherently depends on the availability and quality of external knowledge sources. Specifically, the performance of the pipeline relies on the following: (1) accurate part-of-speech (PoS) tagging, (2) comprehensive and context-relevant thesauri, and (3) sufficient coverage in Wikipedia. These dependencies introduce certain robustness constraints. For example, in low-resource or emerging domains where structured thesauri are not available, or in informal text genres such as social media that include many novel or slang expressions not covered by Wikipedia, the performance gains from our approach may be limited. Similarly, PoS tagging tools may be less reliable on noisy or non-standard input. While our modular design allows selective activation of individual steps depending on the context, future work can

explore adaptive strategies and fallback mechanisms to improve robustness under such conditions.

7. Conclusions

AKE has a more important role in IR and NLP with the increasingly vast amount of digital textual data that modern systems process. In this paper, we aimed to show that an enhanced level of semantic-awareness supported by PoS tagging can improve AKE algorithms. We selected five algorithms as the baseline methods upon experiments comparing several state-of-the-art AKE methods. Then, we used PoS tagging, integrated thesauri, and Wikipedia named entities for improving the baselines. Our experiments on 17 English datasets indicated that the three proposed mechanisms improved the baseline algorithms significantly and consistently.

Author Contributions: Conceptualization, E.A. and S.L.; methodology, E.A., J.R.C.N. and S.L.; software, E.A.; validation, E.A., Y.X. and J.G.; formal analysis, E.A.; investigation, E.A.; writing—original draft preparation, E.A.; writing—review and editing, J.R.C.N. and S.L.; supervision, J.R.C.N. and S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The source code and the complete experimental results of our work are publicly available at https://github.com/altuncu/AKE.

Acknowledgments: We would like to thank Ricardo Campos for clarification and additional information about the YAKE! algorithm. The first author E. Altuncu was supported by funding from the Ministry of National Education, Republic of Türkiye, under grant number MoNE-YLSY-2018.

Conflicts of Interest: The authors declare no competing interest.

References

1. Merrouni, Z.A.; Frikh, B.; Ouhbi, B. Automatic Keyphrase Extraction: A Survey and Trends. *J. Intell. Inf. Syst.* **2020**, *54*, 391–424. https://doi.org/10.1007/s10844-019-00558-9.
2. Gavrilescu, M.; Leon, F.; Minea, A.A. Techniques for Transversal Skill Classification and Relevant Keyword Extraction from Job Advertisements. *Information* **2025**, *16*, 167. https://doi.org/10.3390/info16030167.
3. Papagiannopoulou, E.; Tsoumakas, G. A review of keyphrase extraction. *WIREs Data Min. Knowl. Discov.* **2020**, *10*, e1339. https://doi.org/10.1002/widm.1339.
4. Firoozeh, N.; Nazarenko, A.; Alizon, F.; Daille, B. Keyword extraction: Issues and methods. *Nat. Lang. Eng.* **2020**, *26*, 259–291. https://doi.org/10.1017/S1351324919000457.
5. Campos, R.; Mangaravite, V.; Pasquali, A.; Jorge, A.; Nunes, C.; Jatowt, A. YAKE! Keyword extraction from single documents using multiple local features. *Inf. Sci.* **2020**, *509*, 257–289. https://doi.org/10.1016/j.ins.2019.09.013.
6. El-Beltagy, S.R.; Rafea, A. KP-Miner: A keyphrase extraction system for English and Arabic documents. *Inf. Syst.* **2009**, *34*, 132–144. https://doi.org/10.1016/j.is.2008.05.002.
7. Škrlj, B.; Repar, A.; Pollak, S. RaKUn: Rank-based Keyword Extraction via Unsupervised Learning and Meta Vertex Aggregation. In Proceedings of the 7th International Conference on Statistical Language and Speech Processing (SLSP '19), Ljubljana, Slovenia, 14–16 October 2019; Volume 11816, pp. 311–323. https://doi.org/10.1007/978-3-030-31372-2_26.
8. Ushio, A.; Liberatore, F.; Camacho-Collados, J. Back to the Basics: A Quantitative Analysis of Statistical and Graph-Based Term Weighting Schemes for Keyword Extraction. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP '21), Online, 7–11 November 2021; pp. 8089–8103. https://doi.org/10.18653/v1/2021.emnlp-main.638.
9. Sun, Y.; Qiu, H.; Zheng, Y.; Wang, Z.; Zhang, C. SIFRank: A New Baseline for Unsupervised Keyphrase Extraction Based on Pre-Trained Language Model. *IEEE Access* **2020**, *8*, 10896–10906. https://doi.org/10.1109/ACCESS.2020.2965087.
10. Jones, K.S. A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *J. Doc.* **1972**, *28*, 11–21. https://doi.org/10.1108/eb026526.

11. Brin, S.; Page, L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In Proceedings of the Seventh International World Wide Web Conference (WWW '98), Brisbane, Australia, 14–18 April 1998; Volume 30; pp. 107–117. https://doi.org/10.1016/S0169-7552(98)00110-X.
12. Mihalcea, R.; Tarau, P. TextRank: Bringing Order into Text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004; pp. 404–411.
13. Wan, X.; Xiao, J. Single Document Keyphrase Extraction Using Neighborhood Knowledge. In Proceedings of the 23rd National Conference on Artificial Intelligence, Chicago, IL, USA, 13–17 July 2008; Volume 2, pp. 855–860.
14. Rose, S.; Engel, D.; Cramer, N.; Cowley, W. Automatic Keyword Extraction from Individual Documents. In *Text Mining: Applications and Theory*; Wiley: Hoboken, NJ, USA, 2010; Chapter 1; pp. 1–20. https://doi.org/10.1002/9780470689646.ch1.
15. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**. https://doi.org/10.48550/arXiv.1301.3781.
16. Pennington, J.; Socher, R.; Manning, C.D. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP '14), Doha, Qatar, 25–29 October 2014; pp. 1532–1543. https://doi.org/10.3115/v1/D14-1162.
17. Bennani-Smires, K.; Musat, C.; Hossmann, A.; Baeriswyl, M.; Jaggi, M. Simple Unsupervised Keyphrase Extraction using Sentence Embeddings. In Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL '18), Brussels, Belgium, 31 October–1 November 2018; pp. 221–229. https://doi.org/10.18653/v1/K18-1022.
18. Zhang, L.; Chen, Q.; Wang, W.; Deng, C.; Zhang, S.; Li, B.; Wang, W.; Cao, X. MDERank: A Masked Document Embedding Rank Approach for Unsupervised Keyphrase Extraction. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, 22–27 May 2022; pp. 396–409. https://doi.org/10.18653/v1/2022.findings-acl.34.
19. Rabby, G.; Azad, S.; Mahmud, M.; Zamli, K.Z.; Rahman, M.M. TeKET: A Tree-based Unsupervised Keyphrase Extraction Technique. *Cogn. Comput.* **2020**, *12*, 811–833. https://doi.org/10.1007/s12559-019-09706-3.
20. Liu, Z.; Li, P.; Zheng, Y.; Sun, M. Clustering to Find Exemplar Terms for Keyphrase Extraction. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP '09), Singapore, 6–7 August 2009; pp. 257–266.
21. Witten, I.H.; Paynter, G.W.; Frank, E.; Gutwin, C.; Nevill-Manning, C.G. KEA: Practical Automated Keyphrase Extraction. In *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*; IGI Global: Hershey, PA, USA, 2005; pp. 129–152. https://doi.org/10.4018/978-1-59140-441-5.ch008.
22. Basaldella, M.; Antolli, E.; Serra, G.; Tasso, C. Bidirectional LSTM Recurrent Neural Network for Keyphrase Extraction. In Proceedings of the Digital Libraries and Multimedia Archives; Udine, Italy, 25–26 January 2018; Springer: Berlin/Heidelberg, Germany, pp. 180–187. https://doi.org/10.1007/978-3-319-73165-0_18.
23. Martinc, M.; Škrlj, B.; Pollak, S. TNT-KID: Transformer-based neural tagger for keyword identification. *Nat. Lang. Eng.* **2022**, *28*, 409–448. https://doi.org/10.1017/S1351324921000127.
24. Wang, Y.; Liu, Q.; Qin, C.; Xu, T.; Wang, Y.; Chen, E.; Xiong, H. Exploiting Topic-Based Adversarial Neural Network for Cross-Domain Keyphrase Extraction. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM '18), Singapore, 17–20 November 2018; pp. 597–606. https://doi.org/10.1109/ICDM.2018.00075.
25. Bordoloi, M.; Chatterjee, P.C.; Biswas, S.K.; Purkayastha, B. Keyword extraction using supervised cumulative TextRank. *Multimed. Tools Appl.* **2020**, *79*, 31467–31496. https://doi.org/10.1007/s11042-020-09335-1.
26. Hulth, A. Improved Automatic Keyword Extraction Given More Linguistic Knowledge. In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, Sapporo, Japan, 11–12 July 2003; pp. 216–223.
27. Pay, T. Totally automated keyword extraction. In Proceedings of the 2016 IEEE International Conference on Big Data, Washington, DC, USA, 5–8 December 2016; pp. 3859–3863. https://doi.org/10.1109/BigData.2016.7841059.
28. Zervanou, K. UvT: The UvT Term Extraction System in the Keyphrase Extraction task. In Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval-2010), Uppsala, Sweden, 15–16 July 2010; pp. 194–197.
29. Li, G.; Wang, H. Improved Automatic Keyword Extraction Based on TextRank Using Domain Knowledge. In Proceedings of the Third CCF International Conference on Natural Language Processing and Chinese Computing (NLPCC '14), Shenzhen, China, 5–9 December 2014; Volume 496, pp. 403–413. https://doi.org/10.1007/978-3-662-45924-9_36.
30. Gazendam, L.; Wartena, C.; Brussee, R. Thesaurus Based Term Ranking for Keyword Extraction. In Proceedings of the 2010 Workshops on Database and Expert Systems Applications, Bilbao, Spain, 30 August–3 September 2010; pp. 49–53. https://doi.org/10.1109/DEXA.2010.31.
31. Hulth, A.; Karlgren, J.; Jonsson, A.; Boström, H.; Asker, L. Automatic Keyword Extraction Using Domain Knowledge. In Proceedings of the Computational Linguistics and Intelligent Text Processing: Procedings of the Second International Conference on Intelligent Text Processing and Computational Linguistics (CICLing '01), Hanoi, Vietnam, 18–24 March 2001; Volume 2004, pp. 472–482. https://doi.org/10.1007/3-540-44686-9_47.
32. Medelyan, O.; Witten, I.H. Thesaurus Based Automatic Keyphrase Indexing. In Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries, Chapel Hill, NC, USA, 11–15 June 2006; pp. 296–297. https://doi.org/10.1145/1141753.1141819.

33. Sheoran, A.; Jadhav, G.V.; Sarkar, A. SubModRank: Monotone Submodularity for Opinionated Key-phrase Extraction. In Proceedings of the IEEE 16th International Conference on Semantic Computing (ICSC '22), Laguna Hills, CA, USA, 26–28 January 2022; pp. 159–166. https://doi.org/10.1109/ICSC52841.2022.00032.
34. Shi, T.; Jiao, S.; Hou, J.; Li, M. Improving Keyphrase Extraction Using Wikipedia Semantics. In Proceedings of the 2nd International Symposium on Intelligent Information Technology Application, Shanghai, China, 21–22 December 2008; Volume 2; pp. 42–46. https://doi.org/10.1109/IITA.2008.211.
35. Yu, Y.; Ng, V. WikiRank: Improving Unsupervised Keyphrase Extraction using Background Knowledge. In Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC '18), Miyazaki, Japan, 7–12 May 2018; pp. 3723–3727.
36. Ferragina, P.; Scaiella, U. TAGME: On-the-Fly Annotation of Short Text Fragments (by Wikipedia Entities). In Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM '10), Toronto, ON, Canada, 26–30 October 2010; pp. 1625–1628. https://doi.org/10.1145/1871437.1871689.
37. Papagiannopoulou, E.; Tsoumakas, G. Local Word Vectors Guiding Keyphrase Extraction. *Inf. Process. Manag.* **2018**, *54*, 888–902. https://doi.org/10.1016/j.ipm.2018.06.004.
38. Zesch, T.; Gurevych, I. Approximate Matching for Evaluating Keyphrase Extraction. In Proceedings of the International Conference RANLP-2009, Borovets, Bulgaria, 14–16 September 2009; pp. 484–489.
39. Marujo, L.; Gershman, A.; Carbonell, J.; Frederking, R.; Neto, J.P. Supervised Topical Key Phrase Extraction of News Stories using Crowdsourcing, Light Filtering and Co-reference Normalization. *arXiv* **2013**. https://doi.org/10.48550/arXiv.1306.4886.
40. Medelyan, O.; Frank, E.; Witten, I.H. Human-Competitive Tagging Using Automatic Keyphrase Extraction. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP '09), Singapore, 6–7 August 2009; pp. 1318–1327.
41. Medelyan, O.; Witten, I.H. Domain-Independent Automatic Keyphrase Indexing with Small Training Sets. *J. Am. Soc. Inf. Sci. Technol.* **2008**, *59*, 1026–1040. https://doi.org/10.1002/asi.20790.
42. Das Gollapalli, S.; Caragea, C. Extracting Keyphrases from Research Papers Using Citation Networks. *Proc. AAAI Conf. Artif. Intell.* **2014**, *28*. pp. 1629–1635. https://doi.org/10.1609/aaai.v28i1.8946.
43. Gallina, Y.; Boudin, F.; Daille, B. KPTimes: A Large-Scale Dataset for Keyphrase Generation on News Documents. In Proceedings of the 12th International Conference on Natural Language Generation (INLG '19), Tokyo, Japan, 29 October–1 November 2019; pp. 130–135. https://doi.org/10.18653/v1/W19-8617.
44. Krapivin, M.; Autaeu, A.; Marchese, M. *Large Dataset for Keyphrases Extraction*; Departmental Technical Report DISI-09-055; University of Trento, Trento, Italy, 2009.
45. Nguyen, T.D.; Kan, M.Y. Keyphrase Extraction in Scientific Publications. In Proceedings of the Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers: Proceedings of the 10th International Conference on Asian Digital Libraries (ICADL '07), Hanoi, Vietnam, 10–13 December 2007; Volume 4822, pp. 317–326. https://doi.org/10.1007/978-3-540-77094-7_41.
46. Gay, C.W.; Kayaalp, M.; Aronson, A.R. Semi-Automatic Indexing of Full Text Biomedical Articles. In Proceedings of the 2005 AMIA Symposium. American Medical Informatics Association (AMIA), Washington, DC, USA, 22–26 October 2005; pp. 271–275.
47. Schutz, A.T. Keyphrase Extraction from Single Documents in the Open Domain Exploiting Linguistic and Statistical Methods. Master's thesis, National University of Ireland, Galway, Ireland, 2008.
48. Kim, S.N.; Medelyan, O.; Kan, M.Y.; Baldwin, T. SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles. In Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval-2010), Los Angeles, CA, USA, 15–16 July 2010; pp. 21–26.
49. Augenstein, I.; Das, M.; Riedel, S.; Vikraman, L.; McCallum, A. SemEval 2017 Task 10: ScienceIE-Extracting Keyphrases and Relations from Scientific Publications. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 546–555. https://doi.org/10.18653/v1/S17-2091.
50. Medelyan, O.; Witten, I.H.; Milne, D. Topic Indexing with Wikipedia. In Proceedings of the 2008 AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy, Chicago, IL, USA, 13 July 2008; pp. 19–24.
51. Smadja, F. Retrieving Collocations from Text: Xtract. *Comput. Linguist.* **1993**, *19*, 143–178.
52. Justeson, J.S.; Katz, S.M. Technical Terminology: Some Linguistic Properties and an Algorithm for Identification in Text. *Nat. Lang. Eng.* **1995**, *1*, 9–27. https://doi.org/10.1017/S1351324900000048.
53. Ajallouda, L.; Fagroud, F.Z.; Zellou, A.; Benlahmar, E.H. A Systematic Literature Review of Keyphrases Extraction Approaches. *Int. J. Interact. Mob. Technol. (iJIM)* **2022**, *16*, 31–58. https://doi.org/10.3991/ijim.v16i16.33081.
54. Bird, S. NLTK: The Natural Language Toolkit. In Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, Sydney, Australia, 17–18 July 2006; pp. 69–72. https://doi.org/10.3115/1225403.1225421.
55. Choueka, Y. Looking for Needles in a Haystack or Locating Interesting Collocational Expressions in Large Textual Databases. In Proceedings of the RIAO Conference on User-Oriented Content-Based Text and Image Handling, Cambridge, MA, USA, 21–24 March 1988; pp. 609–623.

56. Boudin, F. PKE: An Open Source Python-Based Keyphrase Extraction Toolkit. In Proceedings of the 26th International Conference on Computational Linguistics: System Demonstrations (COLING '16), Osaka, Japan, 13–16 December 2016; pp. 69–73.
57. Caracciolo, C.; Stellato, A.; Morshed, A.; Johannsen, G.; Rajbhandari, S.; Jaques, Y.; Keizer, J. The AGROVOC Linked Dataset. *Semant. Web* **2013**, *4*, 341–348. https://doi.org/10.3233/SW-130106.
58. Lipscomb, C.E. Medical Subject Headings (MeSH). *Bull. Med. Libr. Assoc.* **2000**, *88*, 265–266.
59. Salatino, A.A.; Thanapalasingam, T.; Mannocci, A.; Osborne, F.; Motta, E. The Computer Science Ontology: A Large-Scale Taxonomy of Research Areas. In Proceedings of the Semantic Web: Proceedings of the 17th International Semantic Web Conference (ISWC '18), Part II, Monterey, CA, USA, 8–12 October 2018; Volume 11137, pp. 187–205. https://doi.org/10.1007/978-3-030-00668-6_12.
60. Osborne, F.; Motta, E. Klink-2: Integrating Multiple Web Sources to Generate Semantic Topic Networks. In Proceedings of the Semantic Web: Proceedings of the 14th International Semantic Web Conference (ISWC '15), Part I, Bethlehem, PA, USA, 11–15 October 2015; Volume 9366, pp. 408–424. https://doi.org/10.1007/978-3-319-25007-6_24.
61. Kempf, A.O.; Neubert, J. The Role of Thesauri in an Open Web: A Case Study of the STW Thesaurus for Economics. *Knowl. Organ.* **2016**, *43*, 160–173. https://doi.org/10.5771/0943-7444-2016-3-160.
62. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
63. Arora, S.; Liang, Y.; Ma, T. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In Proceedings of the 2017 International Conference on Learning Representations (ICLR '17), Toulon, France, 24–26 April 2017; pp. 1–16.
64. Jauhiainen, T.; Lui, M.; Zampieri, M.; Baldwin, T.; Lindén, K. Automatic Language Identification in Texts: A Survey. *J. Artif. Intell. Res.* **2019**, *65*, 675–782. https://doi.org/10.1613/jair.1.11675.