Automatic Keyphrase Extraction : An Overview Of The State Of The Art

Zakariae Alami Merrouni
LTTI Lab
EST, Sidi Mohamed Ben Abdellah University
B.P. 2427, Route d'imouzer,
Fès, Morocco
zakariae.alamimerrouni@usmba.ac.ma

Bouchra Frikh
LTTI Lab
EST, Sidi Mohamed Ben Abdellah University
B.P. 2427, Route d'imouzer,
Fès, Morocco
bfrikh@yahoo.com

Brahim Ouhbi
MODEC Lab
ENSAM, Moulay Ismaïl University,
Marjane II, B.P. 4024,
Meknès, Morocco
ouhbib@yahoo.co.uk

***Abstract*—Keyphrases are useful for a variety of tasks in information retrieval systems and natural language processing, such as text summarization, automatic indexing, clustering/classification, ontology learning and building and conceptualizing particular knowledge domains, etc. However, assigning these keyphrases manually is time consuming and expensive in term of human resources. Therefore, there is a need to automate the task of extracting keyphrases. A wide range of techniques of keyphrase extraction have been proposed, but they are still suffering from the low accuracy rate and poor performance. This paper presents a state of the art of automatic keyphrase extraction approaches to identify their strengths and weaknesses. We also discuss why some techniques perform better than others and how can we improve the task of automatic keyphrase extraction.**

Keywords: *Automatic Keyphrase Extraction; Information Retrieval; Natural Language Processing.*

1 INTRODUCTION

Keyphrases, which can be single keywords or multiword key terms, provide a high level description of a document. Document keyphrases are important for many areas of natural language processing (NLP) and information retrieval (IR) tasks, such as document classification and clustering [1][2], information retrieval systems [3], opinion mining [4], web mining [7][8] , search engines, which supplement full-text indexing and assist users in formulating queries [9] [10], question-answering [12], text summarization [13][15][16], text categorization [17], document content-based recommender systems which help users in discovering information relevant to their previously expressed interests [19]. Furthermore, keyphrase extraction can be used to facilitate the automatic construction of concept maps [20] or ontologies learning and building [41] [48] [56], which allow better understanding of the interconnections and relations between different topics that improve search results [22].

Keyphrase extraction techniques can be also applied on many domains such as medicine [23] [14], Agriculture [25], National security and terrorism [26], social media [27] [28], and law [29], etc.

Assigning keyphrases manually is time consuming and particularly difficult specially in large document sets. Thus, the task of Keyphrase extraction is desirable. It's defined as the problem of automatically extracting expressive, descriptive and important phrases or concepts from a document, which can be as an efficient and practical alternative.

According to some mentioned references, the results of these automatic keyphrase extraction systems have not been satisfactory, and the state-of-the-art performance on keyphrase extraction is still much lower. In this paper, we study and examine the used approaches and techniques of automatic keyphrase extraction (AKPE) and present an overview of the major existing systems by identifying their strengths and weakness points, in order to enhance the first ones and discuss the second ones, these points should be considered in future work of automatic keyphrase extraction techniques in order to achieve a higher accuracy and performance.

This paper is organized as follows: Section 2 presents a review of Automatic Keyphrase Extraction process along with a comparison of realized techniques and their strengths and weakness; In Section 3, we discuss related issues and recommend some solutions, in Section 4, we give a conclusion.

2 AUTOMATIC KEYPHRASE EXTRACTION TASK

2.1 Automatic Keyphrase Extraction Process

A keyphrase is a sequence of one or more words that is considered highly relevant. A keyword is a single word that is highly relevant which can be used as the basis for finding

978-1-5090-0751-6/16/$31.00 ©2016 IEEE

candidate phrases. Thus by identifying the keywords we can identify keyphrases which are more informative than keywords [15].

Keyphrase extraction system consists of 4 main steps: 1) pre-processing; 2) selecting and identifying a list of words/phrases to generate a set of candidate keyphrases using some heuristics; 3) determining which of these candidate keyphrases are correct keyphrases by using supervised or unsupervised approaches; 4) evaluating these extracted keyphrases “Fig. 1,”.

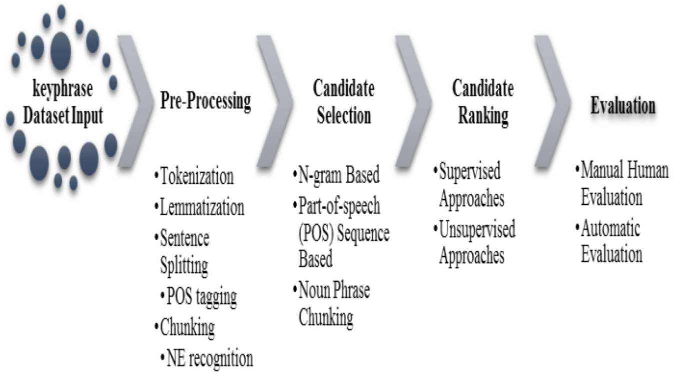

Fig. 1. The Steps of Automatic Keyphrase Extraction Process

2.1.1 Document preprocessing and Candidate phrase identification or selection

The pre-processing task can include formatting the document, sentence segmentation, tokenization, stemming of text, Part-of-Speech tagging, removal of noisy symbols and stop words, then candidate phrases are identified and selected from the full text document. A set of phrases and words is typically extracted as candidate keyphrases using heuristic rules which aimed to reduce the number of false-positive candidates while maintaining the true positives, and keep the number of candidates to a minimum. Existing approaches usually fall into two categories: n-gram based and part-of-speech (POS) sequence based. Typical n-gram based approach was adopted in original systems [30] [6] where the input text is separated according to phrase boundaries (e.g., punctuation marks, and numbers...); limit length (bigram, trigram…) are reserved as candidate phrases; then, some simple rules are applied to filter meaningless subsequences (e.g., candidate phrases cannot begin or end with a stopword) [31] [11]. POS sequence-based approaches are also widely used which allow words with definite part of- speech tags (e.g., adjectives, nouns, verbs) to be candidate keywords using POS tagger [32].

By using these heuristics, several systems obtain a high recall in extracting gold keyphrases. In the case of resulting long candidate list, in [31] [60] [25] [34] [35] different pruning heuristics have been designed in order to limit candidates that are unlikely to be keyphrases.

2.1.2 Keyphrase Extraction Approaches

a) Supervised Approaches

- The Classification Stage

In supervised approaches the Keyphrase extraction is considered as a binary classification problem where each candidate phrase in a document determined as a keyphrase or not [5] [6] [21]. Hence, is machine learning that provide tools to cope this kind of situation. In machine learning terminology, the phrases in a document are “examples” and the learning problem is defining a mapping from the examples to the two classes “Keyphrase” and “not-Keyphrase”. Hence, by training a classifier on documents annotated with keyphrases we can determine whether a candidate phrase is a keyphrase.

Different machine learning approaches are used like bagged C4.5 [5], naive Bayes [21], neural networks [24], SVM [36] maximum entropy [37]. Therefore, several feature sets and classification algorithms give rise to different models:

KEA (1999)

Kea [30] identifies candidate keyphrases using lexical methods, calculates feature values for each candidate and uses a machine-learning algorithm (Naive Bayes) to predict and assign candidates as good keyphrases. The Naïve Bayes model is used on a candidate phrase with feature values *t* (for TF×IDF) and *d* (for distance), two quantities are computed (1):

$$P[yes] = \frac{Y}{Y+N} P_{TF\times IDF}[t|yes] P_{distance}[d|yes] \qquad (1)$$

and a similar expression for P[no], where Y is the number of positive instances in the training files that is, author-identified keyphrases and N is the number of negative instances that is, candidate phrases that are not keyphrases. The overall probability (2) that the candidate phrase is a keyphrase can then be calculated:

$$p = \frac{P[yes]}{P[yes]+P[no]} \qquad (2)$$

However, KEA depends on the training set and may provide poor results when the training set does not fit well the processed documents.

GenEx (2000)

GenEx [6] one of the most known keyphrase extraction techniques, is based on a set of parametrized heuristic rules that are fine-tuned using a genetic algorithm based on a C4.5 decision-tree-like process and has the ability to retain its performance across different domains. However, GenEx approach uses many more attributes among them *distance* but does not use *TF×IDF*.

HUMB (2010)

In HUMB system [44], 3 kind of features have been used, Structural features (position, etc..), content features (Phraseness, Informativeness, Keywordness) and finally the Lexical/Semantic features (Wikipedia keyphraseness, etc.).

As a machine learning, 3 models have been used and combined with boosting and bagging techniques. These models are Decision tree (C4.5), Multi-Layer perceptron (MLP) and Support Vector Machine (SVM). The selected model for the final run was, therefore, bagged decision tree. Good results have been achieved by the HUMB system on different data sets [64]. However, HUMB used the external knowledge bases which are specific to scientific papers.

DPM-index (2014)

Haddoud & al., (2014) [40] Define the document phrase maximality index (DPM-index), a new measure to discriminate overlapping keyphrase candidates in a text document. As an application they developed a supervised learning system that uses 18 statistical features including the DPM-index and five other new features. The results of this system achieved remarkable improvements over other key phrase extraction systems, without using any external knowledge or document structural features [64].

CeKE (2015)

Bulgarov and Caragea, (2015) [63] showed that, in addition to a document textual content and textually-similar neighbors, other informative neighborhoods exist. They have the potential to improve keyphrase extraction and adding the keyphraseness feature (which shows how often a candidate phrase appears as a tag or a keyphrase in the training dataset). The (CeKE + keyphraseness) model outperform other systems (Maui [45], Hulth systems [18]).

- Feature Engineering

Designing a supervised keyphrase extraction system consists of selecting the properties that could distinguish keyphrases from other terms. These properties, called features, which are used to represent an instance for supervised keyphrase extraction and reflect how well a candidate phrase represents the topic and the content of the document.

In the calculating feature step, a number of features are calculated for each candidate phrase to measure and index its importance. Each candidate phrase is characterized by statistical and linguistic properties. They are used for ranking and selecting the final keyphrases. The majority of used features combine 1) frequency statistics ; within a single document and across an entire collection ; 2) semantic similarity among keyphrases (i.e. keyphrase cohesion), popularity of keyphrases among manually assigned sets, 3) heuristics such as locality and the length of phrases [61] [38] [39], and 4) lexical and morphological analysis. Features can be classified according to their origin: phrase-level features, document-level features, corpus-level features and external knowledge-based features.

Generally speaking, there is features which can be used in supervised learning (feature weights rely on priors or statistics attained in the training process, such as tf-idf, Distance of a phrase, Accuracy, F1- Measure, Mutual Information, Term Strength, Chi-Square, Information Gain, and Odds Ratio) or in both supervised and unsupervised learning (features which can be extracted without training like structural rules (tags or positions) or frequency- based information: tf, tf-idf).

Hence, features can be divided into two categories: firstly, Internal-Collection Features: like Statistical and Baseline set features within a single document and across an entire collection; Structural features; Syntactic and morphological features that encode the syntactic patterns of a candidate keyphrase. Secondly, External Resource-Based Features: such as article databanks (Wikipedia, etc.), terminological databases: MeSH, GRISP, etc.; linguistic resources (WordNet, etc.); search engine query logs and search engine results. (see Fig. 2).

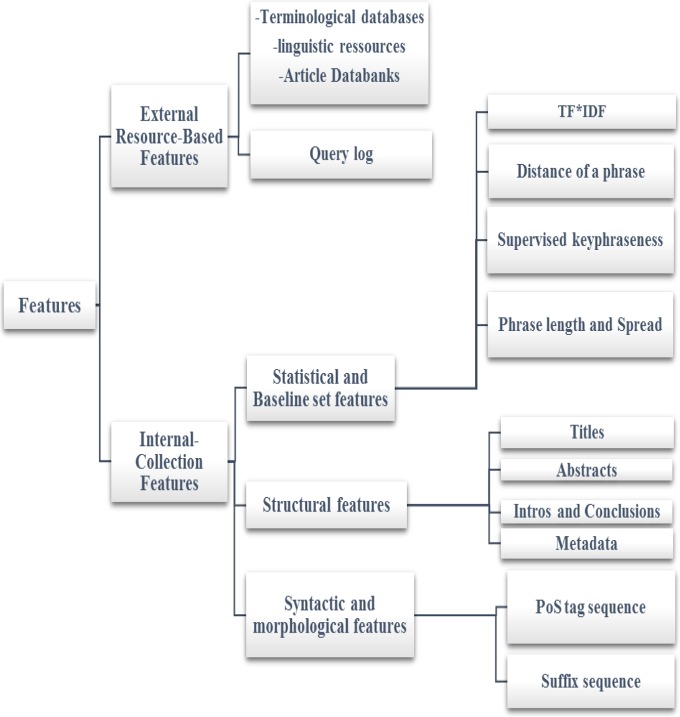

Fig. 2. Baseline Features of Automatic Keyphrase Extraction

b) Unsupervised Approaches

While supervised approaches require a large amount of training data, unsupervised methods can perform without prior knowledge. In unsupervised approaches, keyphrase extraction task is considered as a ranking problem and can be gathered into statistical-based and graph-based approaches. In statistical-based approaches, we usually represent texts as matrices in which the statistical techniques are applied to rank the words by using *tf-idf* term weighting metric [62] [43]. Matsuo and Ishizuka, (2004) [27]; Frikh et al., (2011) [48] present a study that applies CHIR squared test on word co-occurrence distribution. Frantzi et al. (1998) [49] combined linguistics and statistical information to extract technical terms from documents in digital libraries. The more recently developed models are the graphical representation of the text. These graph based methods build a graph from the input documents, and each document is represented as a graph where vertices or nodes represent words, and edges are connected based on either lexical or semantic relations, such as a co-occurrence relation. Nodes or vertex are then ranked using graph centrality measures; such as PageRank and its variants [52]; that assign weights to the node-words reflecting their semantic importance into the text.

The unsupervised systems tended to propose a novel probabilistic model to score candidates, mostly based on simple multiplication of feature values, but also including PageRank family [51] [46] [53], *Topic-Based Clustering* [43] [33] [54], *language modeling [55]*, or *Hybrid* 'statistical and graph based' [58]. Below we review some realized unsupervised systems:

The TextRank (2004)

TextRank Algorithm [51] is one of the most well-known graph based approaches for keyphrase extraction. It represents a document as a graph. Each vertex in the graph corresponds to a word. There is an edge between any two words occurring together.

In TextRank, the in-degree of a vertex is equal to its out-degree, since the graph is undirected. Formally, let *D* denote a document, and *w* denote a word, then $D = \{w_1, w_2, ..., w_n\}$. The weight (3) of a vertex calculated by TextRank is:

$$WS(v_i) = (1-d) + d \times \sum_{v_k \in in(\vartheta_i)} \frac{w_{ij}}{\sum_{v_k \in out(v_j)} w_{jk}} WS(v_j) (3)$$

where w_{ij} is the strength of the connection between two vertices v_i and v_j, and d is the dumping factor, usually set to 0.85 [50].

However, the TextRank does not make full use of the statistical information, such as the length and the position of the phrase, and its best score is achieved when only nouns and adjectives are used to create a uniformly weighted graph for the text under consideration.

KP-Miner (2009)

KPMiner a non-learning, ranking-based system [25], operates on n-grams and uses a modified version of *tf-idf*. KP-Miner boosts the weights of multiword candidates in proportion to the ratio of the frequencies of single word candidates to all candidates.

KPMiner has the advantage of being configurable by users as the rules and heuristics adopted by the system are related to the general nature of documents and keyphrases. Though, KP-Miner did not subtract the overlapping count during its weightage calculation and only considers terms as candidates which occur on their own in the text i.e. surrounded by punctuation marks or stop words. In shorter documents it is more likely that fewer keyphrases would occur in such conditions in the text, causing them to be eliminated early by KP-Miner. A further improvement to the system is involving certain stop words to appear within the produced keyphrases.

SGRank Algorithm Danesh et al., (2015)

Danesh et al., (2015) [58] present a hybrid statistical-graphical algorithm that capitalizes on the heuristics of both two major families of algorithms of unsupervised keyphrase extraction; the statistical and graph-based ones. The SGRank algorithm processes an input document in four stages. They define also a Position of First Occurrence factor (PFO) according to the following formula (4):

$$PFO(t,d) = \log(\frac{cutoffPosition}{p(t,d)}) \quad (4)$$

where p(t,d) is the position of term t's first occurrence in document d, cutoffPosition is set to 3000.

Experiment results; on 2 datasets show that the average performance is considerably better than all of the advanced graph-based algorithms (KP-Miner and TextRank).

Generally, both supervised and unsupervised approaches for keyphrases extraction have some weaknesses and strengths (TABLE I).

TABLE I. STRENGTHS AND WEAKNESSES OF SUPERVISED AND UNSUPERVISED APPROACHES FOR KEYPHRASE EXTRACTION

Supervised Approaches	Unsupervised Approaches
Strengths: (+) More precision. (+) Parameterized heuristic rules. (+) More decision. *Weaknesses:* (-) Need of a gold standard keyphrases and training data to compare and evaluate. (-) Often dependent on the domain. (-) Needs relearn and establish the model every time when a domain changes. (-) Impractical to label training set from time to time by human. (-) Time consuming on training and extraction phase. (-) Use absolute feature values.	*Strengths:* (+) More flexible. (+) Based on unlabeled corpora. (+) Do not exploit any manually tagged corpus or training data. (+) Time efficiency. (+) Good solution for some real-time applications. (+) Simplicity—syntactic representation requires almost no language-specific linguistic processing. (+) Take into account term co-occurrence patterns. *Weaknesses:* (-) The graph based: no guarantee that all the main topics will be represented by the extracted keyphrases. (-) The graph based: no good coverage of the document's.

(+) Strength

(-) weakness

2.1.3. Evaluation

The well-known metrics for evaluating keyphrase extraction algorithms and the output of these keyphrase extraction systems are:

First, the manual evaluation based on human judges by deciding whether the retrieved keyphrases are well representatives of a document's content or not [6] [47]. However, human evaluation of extracted keyphrases is very expensive and time consuming, and it is not appropriate for any kind of parameter tuning.

Second, since exact matching can be excessively strict, researchers have developed a number of systems for automatic evaluation via partial matching. Automatic keyphrase extraction systems have commonly been evaluated using the amount of top N candidates that correctly match the gold-standard keyphrases. This number is then used to calculate the precision, recall and F-score for a keyphrase set. These systems are often used in machine translation and summarization evaluations. However, these systems only offer a partial solution as they can only detect a subset of near-misses to exact match terms [42]. The Automated evaluation relies on matching a set of human annotated gold standard keyphrases with the ranked keyphrases extracted by a certain approach. This matching can also be true or false, depending on whether gold standard keyphrases and the ranked keyphrases extracted are equivalent according to the matching strategy. Automated evaluation relies also on evaluating the global performance of a keyphrase extraction system; which means the full list of matchings. Indeed, keyphrase ranking can also be used as an evaluation metric to distinguish between systems that score the same number of exact matches [57] or to determine which systems rank the most correct keyphrases above incorrect candidate keyphrases [59].

Previously, Precision (P), Recall (R), and F-Measure (F) were used in many studies, at a certain fixed cutoff value. The information retrieval R-precision (R-p) is also used. It is defined as the precision when the number of retrieved documents equals the number of relevant documents in the document Collection. An R-precision of 1.0 is equivalent to perfect keyphrase ranking and perfect recall. That's mean a system that achieve a perfect *Rp* value if it ranks all the keyphrases above the non-keyphrases [59].

3 DISCUSSION AND RECOMMENDATION

By Reviewing the 27 keyphrase extraction techniques (from 1999 to 2015), the precision and recall rates of keyphrases are still as low as 0.40 out of 1.00 (see TABLE II). Thus, by testing, surveying and analyzing some available KPE Systems on their own dataset and other shared dataset, (e.g. SemEval-2010/Task-5) [64]. These systems show worthwhile trends in the different stages of keyphrase extraction: 1) the preprocessing and candidate identification, 2) feature engineering, 3) candidate ranking and evaluation of keyphrases. In the first stage, we notice that the most systems have used either POS-based or n-grams approaches, or both. A challenge a head in this stage, is dealing with the structure of a keyphrase which sometimes can be irregular, it may contain only a single word or a multiword noun phrase or multiple multiword noun phrases connected by prepositions. Moreover, there is a clear obligation to apply pre-processing before to the candidate identification step. In the ranking stage of candidates, some keyphrase extraction systems applied a variety of features: Syntactic/ morphological, statistical and structural ones. Some of these systems used External Resource-Based Features, such as Wikipedia and external corpora terminological databases. However, the information such as document structure, or documents statistics show a better effectiveness of automatic keyphrase extraction. But it is noteworthy that the use of external knowledge bases is computationally expensive than the baseline features.

Until now, El-Beltagy and Rafea, (2009) [25]; Lopez and Romary (2010) [44]; Haddoud and Abdeddaim (2014) [40], show the best performance of these systems based on their own dataset and also on the shared SemEval-2010/Task-5 data [65].

To rank the candidates, supervised systems used learners such as naive Bayes and C4.5 bagged decision trees, maximum entropy, logistic regression, and learn-to-rank classifier based on SVMrank. it's clear that the use of bagged decision trees C4.5 lead to better results. However, in supervised approaches, some keyphrases are positively identified as candidate keyphrase but failed to be ranked at the top for extraction. In unsupervised systems, graph-based methods are limited to a single topic of the document and fail to cover other substantial topics of the document. According to the results of evaluation stage of the supervised and unsupervised techniques, some limitations are raised such as: (1) most of the systems overlook the infrequent relevant keyphrases, and just focus on frequent words in the associated documents (2) redundancy is induced by semantically equivalent keyphrases (3) the global performance of all ranking algorithms drops with an increase in the length or the number of documents (4) frequency alone cannot capture semantic relationships (or context) in the text, and therefore may provide unsatisfactory candidates for keyphrases extraction. (5) the local statistical information is that it does not cover semantic information about the words appearing in a document. (6) Many studied systems have evaluated their approaches on one dataset. An evaluation on many datasets is recommended.

Table. *II* provides a summary of some prominent automatic keyphrase extraction techniques, the used dataset, their features, and the best scores achieved by each technique.

For any future keyphrase extraction system, we recommend: to focus more on preprocessing stage, candidate selection and features weighting process stages.

In the feature engineering stage, we believe that the success of automatic keyphrase extraction depends mainly on the quality, number and variety (statistic, structural and linguistic) of the used features such as the relative position of the phrase, keyphraseness, tf*idf, co-occurrence, length of the phrase candidate, etc. Additionally, despite that the use of external knowledge bases are computationally expensive, using them make the keyphrases more grammatically understandable for the user and increase the precision, recall and exact match rates. In the ranking stage, generally, unsupervised systems are dramatic time saving and have the ability of producing a large high-quality keyphrase list which makes the system more effective for not consumes a lot of computing resources, especially in query refinement and real-time applications. In the evaluation stage we should consider the topics and context of words for better performance on precision and recall in keyphrase extraction and good coverage of each document. Moreover, a semantic evaluation should be engaged to give more accurate of keyphrases.

TABLE II. : CHARACTERISTICS OF SOME MENTIONED KEYPHRASE EXTRACTION TECHNIQUES ALONG WITH THE FEATURES USED BY EACH SYSTEM, AND THE BEST SCORES ACHIEVED AS DESCRIBED IN THEIR PAPERS.

Technique /Contributor	**Dataset**	**Documents/ Source**	**Approach**	**Features**		**Average Match/ Precision Achieved**			
				Internal	***External***	***Top Candidates***	***Precision***	***Recall***	***F-Score***
KEA (Witten et al**.,** 1999) [30]	NZDL	1800 Technical Reports	-Supervised Learning -Classification Problem	- TFxIDF - First occurrence position	-	5	0.300	-	-
						10	-	-	-
						15	0.165	-	-
GenEx (Turney,2000) [6]	-	- 75 Journal Articles -311 Email Messages -266 Web Pages	-Supervised Learning -Classification Problem	-Term Frequency (TF) -First occurrence position - Phrase Length	-	5	0.239	-	0.118
						10	-	-	-
						15	0.128	-	-
Hulth system (Hulth., 2003) [18]	*Inspec*	2000 Paper abstracts	-Supervised Learning -Classification Problem	- POS tag(s); noun phrase (NP) chunks; n-grams; TF ; First occurrence position	-	5	-	-	-
						10	-	-	-
						15	-	-	-
TextRank (Mihalcea and Tarau ,2004) [51]	*Inspec*	500 Paper abstracts	-Unsupervised -Graph based	POS	-	5			
						10	0.142	0.125	0.127
						15			
Single rank (Xiaojun & al,2008) [46]	DUC2001	308 news articles	-Unsupervised -Graph based	-POS tagging -Neighborhood -level saliency score	-	5	0.34	0.25	0.26
						10	0.35	0.28	0.31
						15	0.24	0.43	0.30
KP-Miner (El-Beltagy & Rafea, 2009) [25]	CSTR, NASA, FIPS, Journals, Aliweb	502 documents	-Unsupervised -Non-learning Ranking	-TFxIDF -First occurrence position -Boosting Factor for weight bias	-	7	0.214	0.277	0.241
						15	0.143	0.358	0.205
						20	0.124	0.376	0.186
HUMB (Lopez and Romary, 2010) [44]	SemEval-2010/Task-5(ACM Digital Library,NUS corpus)	156 Technical and Scientific documents	-Supervised	-Position of a term; Position of the first occurrence; Phraseness; Informativeness; Keywordness; Length of the term candidate.	-*Wikipeia keyphraseness* -GRISP based term -GROBID/TEI	5	0.390	0.133	0.198
						10	0.320	0.218	0.259
						15	0.272	0.278	0.275
DPM-index (Haddoud et al., 2014) [40]	SemEval-2010/Task-5	244 scientific papers	-Supervised -Classification problem	- Length n of the n-gram t in words; *tf*; *idf*; *TFxIDF*; First position; First sentence; Head frequency; Average sentence length; Substrings frequencies sum; Generalized Dice coefficient; Maximum likelihood estimate; Kullback–Leibler divergence; Document phrase maximality index; DPM-index cross TFIDF; TFIDF ratio of the term and its main compound; k-Means of the normalized positions (k = 1, 2)	-	5	0.448	0.153	0.228
						10	0.362	0.247	0.294
						15	0.283	0.289	0.286
CeKE (Bulgarov and Caragea, 2015) [63]	CiteSeerx	790 papers published in WWW and KDD	-Supervised -Classification problem	-*tf-idf*; relativePos; POS; inCited; inCiting; citation tf-idf; first position; tf-idf-Over; firstPosUnder; keyphraseness	-	WWW	0.251	0.460	0.322
						KDD	0.254	0.440	0.321

4 CONCLUSION

In this paper, we provided and presented an overview of the state of the art in automatic keyphrase extraction and related work in this area. We outlined each stage of keyphrase extraction process, we also analyzed the performance of the baseline and the recent proposed techniques of this task, and identifies their strengths and drawbacks. We demonstrated that there are still challenges ahead on the task of automatic keyphrase extraction due to the insufficient performance of serval techniques. We believe that, it is essential to focus on semantically and syntactically correct phrase aspects and make sure that the keyphrases are semantically relevant with the document topic and context. Finally, it is necessary to evaluate keyphrase extraction systems on multiple and variant datasets to completely comprehend their strengths and weaknesses.

References

[1] K. M. Hammouda, D. N. MATUTE and M. S. KAMEL. Corephrase: Keyphrase extraction for document clustering. In: *Machine Learning and Data Mining in Pattern Recognition*. Springer Berlin Heidelberg, 2005. p. 265-274.

[2] J. CHOI, J.HAN and T.KIM. Web document clustering by using automatic keyphrase extraction. In: *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops*. IEEE Computer Society, 2007. p. 56-59.

[3] S. JONES and M. S. STAVELEY Phrasier: a system for interactive document retrieval using keyphrases. In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999. p. 160-167.

[4] G. Berend. Opinion expression mining by exploiting keyphrase extraction. *In Proceedings of the 5th International Joint Conference on Natural Language Processing*, 2011.pages 1162–1170.

[5] P. D. Turney, Learning to extract keyphrases from text, Technical Report ERB-1057, *National Research Council, Institute for Information Technology*, 1999.

[6] P. D. Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2000, vol. 2, no 4, p. 303-336.

[7] P.Turney. Coherent keyphrase extraction via web mining. In: *Proceedings of the 20th international joint conference on artificial intelligence*, 2003, pp 434–439. Acapulco, Mexico

[8] K.Yan Lam, M. Chen, J. Tao Sun and H.Zeng. A practical system of keyphrase extraction for web pages. In: *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 2005. p. 277-278.

[9] C. Gutwin, G. Paynter, I. Witten, Craig Nevill-Manning and Eibe Frank. Improving browsing in digital libraries with keyphrase indexes. *Decision Support Systems Journal,* 1999, vol. 27, no 1, p. 81-104.

[10] Z.Gong and Q. Liu. Improving keyword based web image search with visual feature distribution and term expansion. *Knowledge and Information Systems*, 2009, vol. 21, no 1, p. 113-132.

[11] F. Liu, D. Pennell, F. Liu and Y. Liu. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In: *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*. Association for Computational Linguistics, 2009. p. 620-628.

[12] D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Girju, R. Goodrum and V. Rus. The structure and performance of an open-domain question answering system. In: Proceedings *of the 38th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2000. p. 563-570.

[13] H. 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*. ACM, 2002. p. 113-120.

[14] W. Liu, B.C. Chung, R. Wang and *N.* Morlet. A genetic algorithm enabled ensemble for unsupervised medical term extraction from clinical letters. *Health information science and systems*, 2015, vol. 3, no 1, p. 1-14.

[15] Y. Zhang, N. Z-Heywood, and E. Milios. World Wide Web site summarization. *Web Intelligence and Agent Systems*. 2004, 2:39–53.

[16] E. D'avanzo and B. Magnini. A keyphrase-based approach to summarization: the lake system at duc-2005. In: *Proceedings of DUC*. 2005.

[17] A. Hulth and B. B. Megyesi. A study on automatically extracted keywords in text categorization. In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2006. p. 537-544.

[18] A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. In: *Proceedings of the 2003 conference on Empirical methods in natural language processing*. Association for Computational Linguistics, 2003. p. 216-223.

[19] F. Ferrara, N. Pudota and C. Tasso. A keyphrase-based paper recommender system. In: *Digital Libraries and Archives*. Springer Berlin Heidelberg, 2011. p. 14-25.

[20] D. B. Leake, A. Maguitman, and T. Reichherzer. 2003.Topic Extraction and Extension to Support Concept Mapping. In *FLAIRS Conference*, pages 325-329.

[21] E. Frank, G. W. PaynteR, I. H. Witten, C. Gutwin and C.G. Nevill-Manning. Domain-specific keyphrase extraction. Proceedings *of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99), pp. 668-673. California: Morgan Kaufmann*,1999b.

[22] N. Do and L. Ho. Domain-specific keyphrase extraction and near-duplicate article detection based on ontology. In: *Computing & Communication Technologies-Research, Innovation, and Vision for the Future (RIVF), 2015 IEEE RIVF International Conference on*. IEEE, 2015. p. 123-126.

[23] K. Sarkar. A hybrid approach to extract keyphrases from medical documents. *arXiv preprint arXiv:1303.1441*, 2013.

[24] K. Sarkar, M. Nasipuri and S. GHOSE. A new approach to keyphrase extraction using neural networks. *arXiv preprint arXiv:1004.3274*, 2010.

[25] S. R. El-Beltagy and A. Rafea. KP-Miner: A keyphrase extraction system for English and Arabic documents. *Information Systems*, 2009, vol. 34, no 1, p. 132-144.

[26] Y. Elovici, B. Shapira, M. Last, O. Zaafrany, M. Friedman, M. Schneider and A. Kandel. Detection of access to terror-related Web sites using an Advanced Terror Detection System (ATDS). *Journal of the American society for information science and technology*, 2010, vol. 61, no 2, p. 405-418.

[27] Y. Matsuo, J. Mori, M. Hamasaki, T. Nishimuraa, H. Takedab, K. Hasidaa and M. Ishizukab. POLYPHONET: an advanced social network extraction system from the web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2007, vol. 5, no 4, p. 262-278.

[28] J. Mori, M. Ishizuka and Y. Matsuo. Extracting Keyphrases to Represent Relations in Social Networks from Web. In: *IJCAI*. 2007. p. 2820-2827.

[29] M. Jungiewicz and M. Łopuszyński. Unsupervised keyword extraction from Polish legal texts. In: *Advances in Natural Language Processing*. Springer International Publishing, 2014. p. 65-70.

[30] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin and G. Nevill-Manning. KEA: Practical automatic keyphrase extraction. In: *Proceedings of the fourth ACM conference on Digital libraries*. ACM, 1999. p. 254-255.

[31] C. Huang, Y. TIAN, Z. ZHOU, C. X. Ling and T. Huang. Keyphrase extraction using semantic networks structure analysis. In: *Data Mining, 2006. ICDM'06. Sixth International Conference on*. IEEE, 2006. p. 275-284.

[32] K. Barker and N. Cornacchia. Using noun phrase heads to extract document keyphrases. In: *Advances in Artificial Intelligence*. Springer Berlin Heidelberg, 2000. p. 40-52.

[33] M. Grineva, M. Grinev, and D. Lizorkin. Extracting key terms from noisy and multi theme documents. In *Proceedings of the 18th International Conference on World Wide Web*, 2009, pages 661–670.

[34] W. You, D. Fontaine and J. P. Barthès. Automatic keyphrase extraction with a refined candidate set. In: *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on*. IET, 2009. p. 576-579.

[35] David Newman, Nagendra Koilada, Jey Han Lau and Timothy Baldwin. Bayesian Text Segmentation for Index Term Identification and Keyphrase Extraction. In: *COLING*. 2012. p. 2077-2092.

[36] Xin Jiang, Yunhua Hu and Hang Li. A ranking approach to keyphrase extraction. In: *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. ACM*, 2009. p. 756-757.

[37] 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*. ACM, 2006. p. 213-222.

[38] T. D. Nguyen and M-Y. Kan. Keyphrase extraction in scientific publications. In: *Proceedings of the 10th international conference on Asian digital libraries: looking back 10 years and forging new frontiers*, December 10-13, 2007, Hanoi, Vietnam. pp 317–326.

[39] O. Medelyan, E. Frank, and I. H. Witten. Human-competitive tagging using automatic keyphrase extraction. In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*. Association for Computational linguistics, 2009. p. 1318-1327.

[40] M. Haddoud and S. Abdeddaïm. Accurate keyphrase extraction by discriminating overlapping phrases. *J. Information Science*,2014, 40(4), 488–500.

[41] B. Fortuna, M. Grobelnik, and D. Mladenić. 2006.Semi-automatic data-driven ontology construction system. In Proceedings of the 9th International multiconference information society, pages 223–226.

[42] S. N. Kim, O. Medelyan, M-Y. Kane and T. Baldwin. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. ACL, 2010. p. 21-26.

[43] Z. Liu, P. Li, Y. Zheng, M. Sun. Clustering to find exemplar terms for keyphrase extraction. In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 2009a. p. 257-266.

[44] P. Lopez and L. Romary. HUMB: Automatic key term extraction from scientific articles in GROBID. In: *Proceedings of the 5th international workshop on semantic evaluation*. Association for Computational Linguistics, 2010. p. 248-251.

[45] O. Medelyan and I. H. Witten. Thesaurus based automatic keyphrase indexing. In: *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*. ACM, 2006. p. 296-297.

[46] X. Wan and J. Xiao. Single document keyphrase extraction using neighborhood knowledge. In: *Proceedings of the 23rd international conference on artificial intelligence*,2008. Chicago, pp 855–860

[47] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificiel Intelligence Tools*, 13(01):157-169, 2004.

[48] B. Frikh, A. S. Djaanfar and B. Ouhbi. A New Methodology for Domain Ontology Construction from The Web. *International Journal on Artificial Intelligence Tools*, 2011, vol. 20, no 06, p. 1157-1170.

[49] K. T. Frantzi, S. Ananiadou and J. Tsujii. The c-value/nc-value method of automatic recognition for multi-word terms. In: *Research and advanced technology for digital libraries*. Springer Berlin Heidelberg, 1998. p. 585-604.

[50] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In*: Seventh International World-Wide Web Conference (WWW 1998), April 14-18, 1998, Brisbane, Australia.*

[51] R. Mihalcea and P. Tarau. TextRank: Bringing Order into Texts. In Proceedings *of EMNLP,* 2004, pages 404-411.

[52] L. Page, S. Brin, R. Motwani and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. *Technical report, Stanford University*, 1998.

[53] S. Das Gollapalli and C. Caragea. Extracting keyphrases from research papers using citation networks. In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. 2014. p. 1629-1635.

[54] A. Bougouin, F. Boudin and B. Daille. Topicrank: Graph-based topic ranking for keyphrase extraction. In: *International Joint Conference on Natural Language Processing (IJCNLP)*. 2013. p. 543-551.

[55] T. Tomokiyo and M. Hurst. A language model approach to keyphrase extraction. In: *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, 2003. pages 33–40.

[56] O. EL Idrissi, B. Frikh and B. Ouhbi. HCHIRSIMEX: An extended method for domain ontology learning based on conditional mutual information. In: *Information Science and Technology (CIST), 2014 Third IEEE International Colloquium in*. IEEE, 2014. p. 91-95.

[57] Z. LIU, W. HUANG, Y. ZHENG and M. Sun. Automatic keyphrase extraction via topic decomposition. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010. p. 366-376.

[58] S. Danesh, T. Sumner, and J. H. Martin. SGRank: Combining Statistical and Graphical Methods to Improve the State of the Art in Unsupervised Keyphrase Extraction. *Lexical and Computational Semantics (* SEM 2015)*, 2015, p. 117.

[59] T. Zesch and I. Gurevych. Approximate matching for evaluating keyphrase extraction. *In Proceedings of the International Conference on Recent Advances in Natural Language Processing 2009*, pages 484–489.

[60] N. Kumar and K. Srinathan. Automatic keyphrase extraction from scientific documents using N-gram filtration technique. In: *Proceedings of the eighth ACM symposium on Document engineering*. ACM, 2008. p. 199-208.

[61] T. D. Nguyen and M-T. Luong. WINGNUS: keyphrase extraction utilizing document logical structure. In: *Proceeding of the 5th international workshop on semantic evaluation*. 2010, ACL, Uppsala, pp 166–169

[62] K. Zhang, H. Xu, J. Tang, J. Li. Keyword extraction using support vector machine. In: *Proceedings of the 7th international conference on web-age information management*, pp 86–96Hong Kong, 2006.

[63] F. Bulgarov and C. Caragea. A Comparison of Supervised Keyphrase Extraction Models. In: *Proceedings of the 24th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 2015. p. 13-14.

[64] Automatic Keyphrase Extraction from Scientific Articles. Task #5 of the 5th workshop on semantic evaluation, 2005. http://semeval2.fbk.eu/semeval2.php?location=tasks&taskid=6

[65] The SemEval-2010 dataset. http://semeval2.fbk.eu/semeval2.php?location=data