VIT-AP UNIVERSITY

NAME : Indu Madamanchi

REGNO : 21BCE9005

PROFFESOR NAME : Dr.SAGAR DHANRAJ PANDE

TOPIC : TERM EXTRACTION USING NATURAL LANGUAGE PROCESSING

TERM EXTRACTION USING
NATURAL LANGUAGE PROCESSING

Abstract

Terminology extraction, shortly called as term extraction, is a subset of information extraction [1]. The main goal of term extraction is to extract required keywords or key phrases from any corpus automatically [1]. Term extraction is used mostly in Computer Science field, especially in Natural Language Processing and Information Retrieval [11]. It can also be used for query refinement, index generation, author assistance, text summarization, etc. This paper provides various techniques like state of art, frequency based approaches, unsupervised automated domain terminology extraction Using chunking, automatic term extraction using part-of-speech features, position rank algorithm, [21] entropy-based term weighting schemes for text categorization, domain specific text processing etc[25]. used for term extraction. In the modern world of technology and commerce, clear terminology is the key to success and is therefore very essential for the growth of competitive organizations [3].

Key words: Terminology extraction [3], Natural language processing, Query refinement, Index generation, Chunking, Text processing.

1. Introduction

In creative writing, authors choose different synonymous words to describe the same thing in order to make the text more vivid. However, this leads to uncertainty in technical writing [13]. For example, a technician following assembly language instructions can’t refer to the exact same part with many different names as it is more confusing and it might increase chances of getting errors [3]. Therefore, a better standardized terminology is highly preferred and recommended in the domain, and tools which can identify terminology to assist a writer in creating much better documents would be appreciated [22]. It might as well benefit assessors and reviewers of similar documents as well as technical proposals [3].

With the developing preservation of foreign money in contemporary-day tech fields, it's far very useful for customers to get admission to a technique of without problems extracting a part of an index that consists of the important thing phrases worried withinside the document [3]. Though substantial efforts are invested in ATE (Automatic Term Extraction), its overall performance is low [3]. Major elements that affecting ATE overall performance measures encompass the dearth of structural consistency (for instance, loss of a described index or abstract), the presence of uncorrelated subjects withinside the identical text, period of files and subject matter changes [17].

Unsupervised algorithms used for domain term extraction aren’t trained and labelled on the corpus hence they do not `contain pre-defined dictionaries or rules [1]. They usually use statistical data from text or document. Lot of these algorithms can be applied to any text datasets as they use stop word lists [1]. The standard unsupervised automatic term extraction (ATE) pipeline includes :

• **Simple Rules**: use POS tagging or chunking for extracting Noun phrases used in multi-term extraction.

• **Naive counting**: It Counts how many times a term appears in the document or corpus.

• **Pre-processing**: Removes punctuations and similar words like stop words from text.

• **Candidate generation and scoring**: It uses ranking algorithms and statistical measures to create possible sets of domain terms

• **Final set**: It considers top N keywords as the output after arranging the ranked terms in decreasing order based on scores.

Many schemes were developed till now to represent the text by using number systems which capture semantics behind the symbols that are being used in the text. The new found ability of representing text in numeric forms increased the capability to investigate various problems like key-phrase extraction, keyword extraction, topic modelling, document clustering, document summarization and many more in text processing domain.

A single isolated term or token does not carry much information with it, so the human brain processes it with reference to its preceding tokens or terms [22]. Hence the main objective of the text processing algorithm is to see the tokens in the form of groups rather than in isolation form. Hence, the text processing systems should have similar techniques to imitate human brain's interpretation skills in text processing.

Several algorithms were developed for extracting terms in order to improve the ease of accessing text files or documents for computers. However, not all algorithms were good enough to provide required level of accuracy. All the techniques and their efficiencies are discussed following the introduction.

2. Literature Survey:

One of the first surveys on term extraction analyses two directions: term recognition itself and automatic indexing. The survey mainly focuses on the TF-IDF methods. The authors of the paper are the first to introduce the features of the term—term hood (relevance of the term to domain) and unit hood (relation between words in multi-word terms) and study term extraction methods according to the point which is an attribute of the corresponding method [4]. The survey also classifies two classes of methods that are linguistic and statistical.

The research paper "Position Rank model: An Unsupervised Approach to term Extraction” is by Xiaojun Wan and Jianguo. Xiaojun suggested an unsupervised approach to key word and key phrase extraction from the scholarly documents. The paper provides an overview of previous works in the field of term extraction and point outs various tools and methods which have been used for key word and key phrase extraction. The authors of the paper begin by discussing the importance of terminology extraction in NLP and information retrieval. They explain how key words and key phrases helps in better understanding of the content in a document and to retrieve or extract relevant information. They also focus on the challenges involved in key word extraction, including the need to extract important terms that reflects the main concepts and highlights of a document.

The paper continues to describe the Position Rank model in term extraction, which is a graph-based approach to extract key terms. The authors explain how the graph-based model represents the given document as a graph, in which edges represent the co-occurrence of words and nodes represent the words [8]. The graph-based model uses a position-sensitive approach for ranking the significance of each word in the graph and then to identify the key words depending upon their position in the document. The authors then explain the Position Rank model tested on several benchmark datasets for term extraction. They compare their graph-based model with various state-of-the-art methods [23], including unsupervised as well as supervised approaches. The results after tests depict that the Position Rank algorithm is better than other models and it achieves state-of-the-art performance on all the datasets [18].

The paper finally concludes with advantages and disadvantages of Position Rank model. The unsupervised nature of the model, that makes it applicable to most of the documents without any need for training data is specially highlighted by the authors. They also state that the Position rank model may not work efficiently for or text with highly technical or specialized language. Overall, the paper provides an overview of different applications of graph-based approaches to term extraction and depicts the efficiency of the Position Rank model.

The paper "Term extraction rules based on the part-of-speech hierarchy" published by Kang et al. is a rule-based approach used for term extraction from the documents or text using a part-of-speech (POS) hierarchy. The motive of the method is to control the limitations of traditional methods which depend on a fixed set of stop words and which require manual tuning for each specific domain. The authors use a hierarchical approach to Part of speech tagging (POS), that involves classifying each part of speech tag into separate levels of accuracy. This structure allows for more flexible control over the identification of keywords and key phrases. The suggested method initially extracts all candidate phrases consisting of two or more consecutive content words. Then, it applies a set of rules depending on the hierarchical POS tags assigned to filter the non-keyword phrases. The authors tested the method on two datasets: a set of scientific papers and a collection of news articles. The results depict that this method outperforms various baseline methods in terms of accuracy [20], recall, and F1 score. The method is shown to be robust covering different domains and languages, that indicates its potential for a broad range of applicability.

Overall, the paper provides a novel approach for term extraction based on a hierarchical POS tagging system. The suggested method shows promising results and also offers an edge over traditional methods, that include increased flexibility and adaptability to various domains. However, the method is restricted to extracting multi-word phrases and may not be as effective for identifying single-word keywords. Furthermore, the proposed method requires the creation of a domain-specific POS hierarchy, which is time-consuming and labour - intensive.

The authors of the article "Entropy-based term weighting schemes for text categorization" are Tao Wang, Yi Cai, Ho-fung Leung, Haoran Xie, and Qing Li. A document is represented by utilising a vector of terms in the Vector Space Model (VSM) [2], which has been frequently utilised in text categorization. In order to enhance the performance of the vector space model, various term weighting strategies were developed because different terms contribute to a document's semantics in varying degrees [2]. Additional research demonstrates that the effectiveness of any term weighting method frequently varies depending on different text categorization tasks, however the mechanism underlying the performance volatility of a scheme is still unclear [24]. Additionally, currently implemented algorithms often weight a term with respect to a specific category locally, without looking at how often a term appears globally across all categories in a corpus. The authors first look at the benefits and drawbacks of word weighting algorithms that are already in use for text categorization [25]. They also look at why some theoretically good approaches, including information gain and the chi-square test, perform poorly in actual tests. The authors then offer a number of entropy-based phrase weighting algorithms to determine a term's ability to differentiate between categories of text [2]. The suggested term weighting schemes routinely outperform the state-of-the-art systems in further testing on five different datasets. Additionally, their findings offer new insight into creating a term weighting scheme that is efficient for text categorization tasks [25].

3. Overview:

Term extraction is the process of extracting relevant key terms or key phrases from text data using Natural Language Processing (NLP) techniques [15]. There are several techniques used in term extraction, including:

3.1.Part-of-speech (POS) tagging:

POS tagging is a technique used to recognise the part of speech of every word in a sentence [5]. It is useful in term extraction because it can identify nouns, adjectives, and verbs, which are often the most important parts of a term.

3.2.Named entity recognition (NER):

NER is a method for locating and categorising named entities in text, such as individuals, groups, and places [5]. It may detect terms that are distinctive to a given domain, making it valuable for term extraction.

3.3.Text segmentation:

Text segmentation is the process of segmenting a text into number of smaller units [14], such as sentences or paragraphs. It is useful in term extraction because it can help identify multi-word terms that occur within a sentence or across multiple sentences.

3.4.Co-occurrence analysis:

Co-occurrence analysis is a technique used to identify words that frequently co-occur with each other in a text. It is useful in term extraction because it can help identify terms that are frequently used together.

3.5.Frequency analysis:

Frequency analysis is a technique used to identify the most frequently occurring words or phrases in a text. It is useful in term extraction because it can help identify the most important terms in a text.

3.6.Term weighting:

Term weighting is a technique used to assign a weight to each term in a text based on its significance in the context [21]. It is very useful in term extraction as it can help identify the most important terms in a text based on their relevance to a particular domain or topic.

3.7.Machine learning:

Machine learning algorithms can be trained on a corpus of text data to identify and extract relevant terms automatically. These algorithms can learn to identify important features of terms, such as their frequency, co-occurrence, and context, and use this information to extract relevant terms from new text data.

Overall, term extraction is a complex task that requires a combination of these techniques and others, depending on the specific requirements of the task.

The two main algorithms discussed in this paper are term extraction using frequency-based approaches and position rank algorithm for text categorization.

4. Page Rank Algorithm:

The model described in the paper is an unsupervised, model that considers both the frequency of a word and the position where it occurs to calculate a biased PageRank for each and every possible candidate phrase or word which can be considered as a key word [1]. For each word, a weight is computed based on its position and later a preference is assigned to each word accordingly.

From the target document for which the keywords are to be extracted, using the NLP toolkit we select only the nouns and adjectives as candidate words. Then an undirected graph G, with vertices or nodes V and edges E, is constructed such that each node is a candidate word. Two nodes are connected by an edge only if the corresponding candidate words occur within a preset window size *w* of contiguous token of words in the document. Then weight of the positive edge is calculated depending upon the co-occurrence of two candidate phrases or words taken [17].

An adjacency matrix M is also constructed from the graph, where its entries are the weights of two nodes and if there is no edge between two nodes, then zero is taken as the entry. The PageRank of each node is then computed in a recursive fashion by adding the normalized scores of the nodes to which is connected. To prevent the PageRank from getting into endless loops, a factor α is added in order to allow to shift the operation towards another node present in the graph [1].

The idea here is that a higher weight is given to those candidate words that occur in the beginning of the document and are more frequent. After calculation of the weights, candidate phrases are formed [1]. Candidate terms that are contiguous in the target corpus or document are grouped as phrases if they also match the “adjective + noun” format. The score for each phrase is calculated and the phrases with the top score are given as the output which are the predicted key phrases for the document.

4.1.Experiments and Results:

To evaluate the performance of Position Rank, the same experiment was carried out on three datasets consisting of various research papers. Here, key phrases are extracted from the title and the abstract of the document and the key phrases given by the author are considered for measuring the overall performance of the model [24].

The Mean Reciprocal Rank (MRR) is one of the evaluation metrics employed by the method. It is determined by the first correct prediction's averaged ranking and is described as follows [4]:

where D is the collection of the documents and rd is the position at which document d's initial correct key word was discovered.

Here, "s" denotes the vector of PageRank scores and M denotes adjacency matrix.

$$s(t+1) = \widetilde{M}.s(t)$$

For matrix M, $\widetilde{M}$ is the normalized form for all m_{ij} belongs to $\widetilde{M}$.

$$m_{ij} = \begin{cases} m_{ij}/\sum_{j=1}^{|v|} m_{ij} & if \sum_{j=1}^{|v|} m_{ij} \neq 0 \\ 0 & otherwise \end{cases}$$

A damping factor α is added in order to ensure that the PageRank does not keep looping into cycles present in the graph [4] .

$$S = \alpha.\widetilde{M}.S + (1-\alpha).\tilde{p}$$

Prior to applying any filters, we then weigh each of the candidate word according to its inverse location in the text. If a word appears more than once in the text, all of its position weights are added [1].

$$\tilde{p} = \left[\frac{p_2}{p_1+p_2+\ldots\ldots+p_{|v|}}, \frac{p_2}{p_1+p_2+\ldots\ldots+p_{|v|}}, \ldots\ldots\ldots, \frac{p_{|v|}}{p_1+p_2+\ldots\ldots+p_{|v|}}\right] [13]$$

Finally, the page rank score of a vertex can be obtained by computing the following equation recursively.

$$S = (1-\alpha).\tilde{p}_i + \alpha . \textstyle\sum_{v_j \in Adj(v_i)} \frac{w_{ji}}{O\,(v_j)} S(v_j)$$

Table1: Position Rank vs baselines in terms of accuracy, F1-score and Recall. Best results are highlighted in blue colour [1]:

dataset	Unsupervis--ed technique	Top-2			Top-4			Top-6			Top-8		
		P%	R%	F1%	P%	R%	F1%	P%	R%	F1%	P%	R%	F1%
KDD	Position Rank	11.1	5.6	7.3	10.8	11.1	10.6	9.8	15.3	11.6	9.2	18.9	12.1
	Position Rank-fp	10.3	5.3	6.8	10.2	10.4	10.0	9.1	13.8	10.9	8.6	17.2	11.3
	TF-IDF	10.5	5.2	6.8	9.6	9.7	9.4	9.2	13.8	10.7	8.7	17.4	11.3
	Text Rank	8.1	4.0	5.3	8.3	8.5	8.1	8.1	12.3	9.4	7.6	15.3	9.8
	Single Rank	9.1	4.6	6.0	9.3	9.4	9.0	9.0	13.1	10.1	8.1	16.4	10.6
	Expand Rank	10.3	5.5	6.9	10.4	10.7	10.1	10.1	14.5	10.9	8.4	17.5	11.0
	TPR	9.3	4.8	6.2	9.1	9.3	8.9	8.9	13.4	10.3	8.0	16.2	10.4
WWW	Position Rank	11.3	5.3	7.0	11.3	10.5	10.5	10.8	14.9	12.1	9.9	18.1	12.3
	Position Rank-fp	9.6	4.5	6.0	10.3	9.6	9.6	10.1	13.8	11.2	9.4	17.2	11.7
	TF-IDF	9.5	4.5	5.9	10.0	9.3	9.3	9.6	13.3	10.7	9.1	16.8	11.4
	Text Rank	7.7	3.7	4.8	8.6	7.9	8.0	8.1	12.3	9.8	8.2	15.2	10.2
	Single Rank	9.1	4.2	5.6	9.6	8.9	8.9	9.3	13.0	10.5	8.8	16.3	11.0
	Expand Rank	10.4	5.3	6.7	10.4	10.6	10.1	9.5	14.7	11.2	8.6	17.7	11.2
	TPR	8.8	4.2	5.5	9.6	8.9	8.9	9.5	13.2	10.7	9.0	16.5	11.2
Nguyen	Position Rank	10.5	5.8	7.3	10.6	11.4	10.7	11.0	17.2	13.0	10.2	21.1	13.5
	Position Rank-fp	10.0	5.4	6.8	10.4	11.1	10.5	11.2	17.4	13.2	10.1	21.2	13.3
	TF-IDF	7.3	4.0	5.0	9.5	10.3	9.6	9.1	14.4	10.9	8.9	18.9	11.8
	Text Rank	6.3	3.6	4.5	7.4	7.4	7.2	7.8	11.9	9.1	7.2	14.8	9.4
	Single Rank	9.0	5.2	6.4	9.5	9.9	9.4	9.2	14.5	11.0	8.9	18.3	11.6
	Expand Rank	9.5	5.3	6.6	9.5	10.2	9.5	9.1	14.4	10.8	8.7	18.3	11.4
	TPR	8.7	4.9	6.1	9.1	9.5	9.0	8.8	13.8	10.5	8.8	18.0	11.5

5. Frequency based approaches:

A corpus of text containing the terms that must be retrieved is first gathered for term extraction utilising frequency-based techniques.

After that, the text is pre-processed by having stop words, unnecessary punctuation, and other noise removed. The text may also need to be normalised by making everything lowercase and eliminating any numbers or special characters.

After that, the text is tokenized by being divided into separate words or n-grams (contiguous groups of n words). A straightforward dictionary data structure, where the keys are the tokens and the values are the associated frequencies, can be used to count the frequency of each token in the corpus. The tokens that are either too uncommon or too prevalent to be taken into account as keywords are filtered out and ignored. For instance, we might want to eliminate tokens that occur more frequently than a particular proportion of the total number of tokens or fewer times than a specific threshold.

The remaining tokens are then sorted according to their frequency or another measure, such as tf-idf (term frequency-inverse document frequency) [9], which considers both the token's frequency in the corpus and rarity within the entire collection of documents. Finally, the top-ranked tokens are selected as extracted terms.

Here are the equations used in frequency-based approaches for term extraction:

Term Frequency (TF) = n/N

n = Number of times a term appears in the document

N = Total number of terms in document [6]

Inverse Document Frequency (**IDF**) = log_e (D/d)

D = Total number of documents [6]

d = Number of documents containing the term [6]

TF-IDF = TF * IDF [6]

Where:

TF is relative frequency of a term in given document [16]

IDF measures the importance of a term in the corpus by calculating the logarithmic ratio of total number of documents to the number of documents containing the given term

TF-IDF is the product of TF and IDF, indicating the significance of a term in a specific document and across the corpus as a whole.

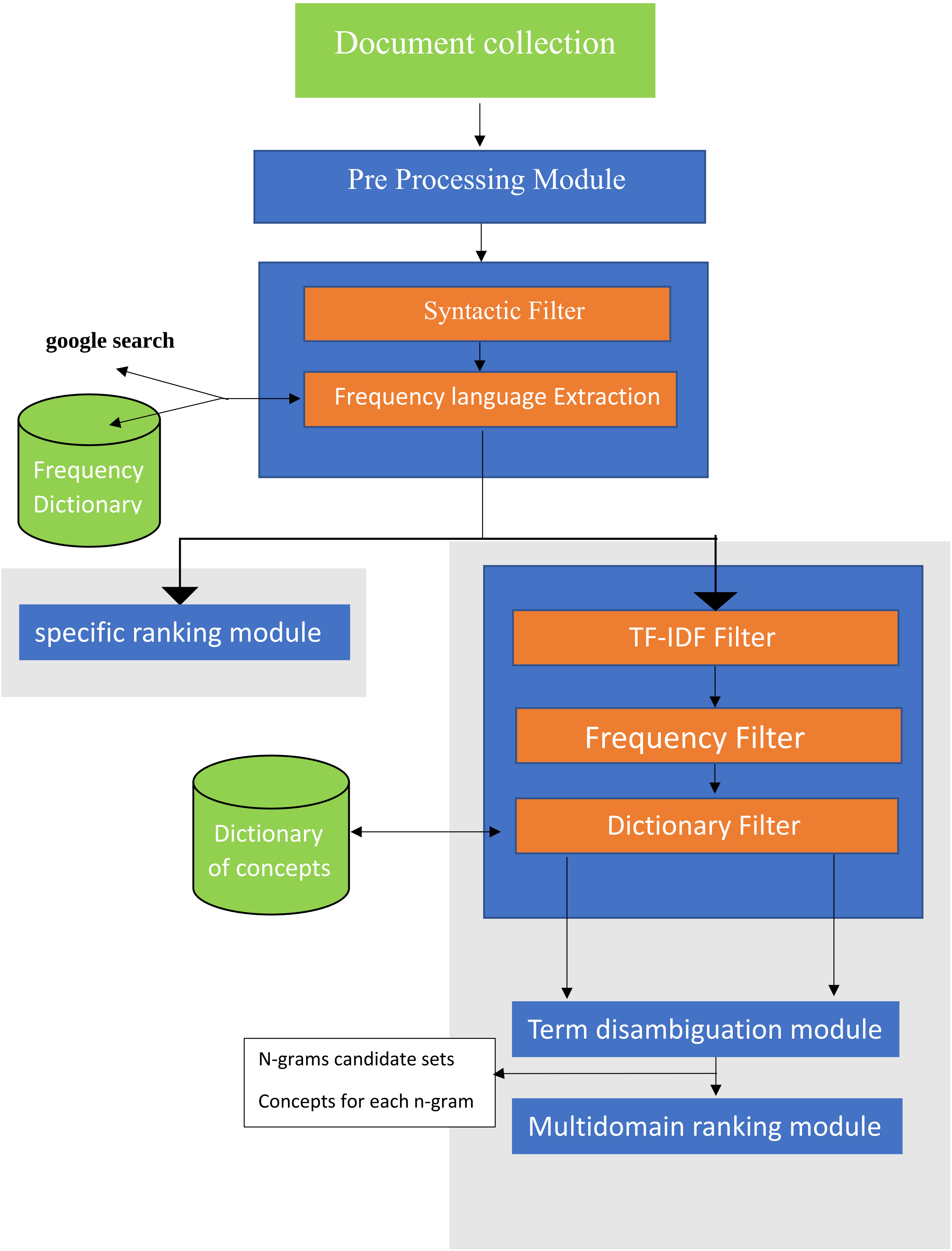
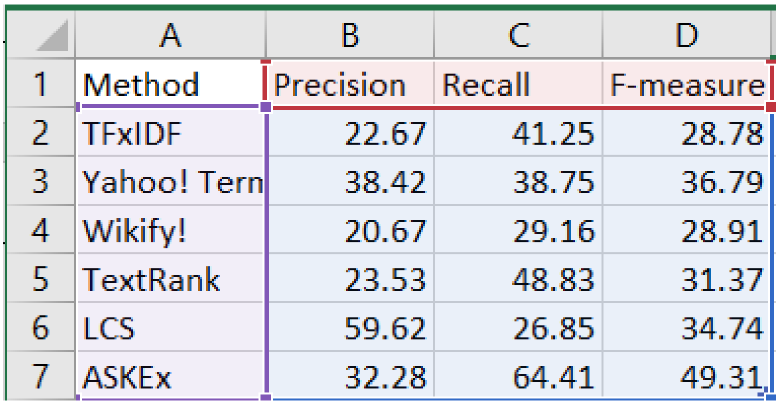


Figure 1: Flow chart describing System Architecture

Table2: Key term extraction results on scientific paper collection[19]



	A	B	C	D
1	Method	Precision	Recall	F-measure
2	TFxIDF	22.67	41.25	28.78
3	Yahoo! Tern	38.42	38.75	36.79
4	Wikify!	20.67	29.16	28.91
5	TextRank	23.53	48.83	31.37
6	LCS	59.62	26.85	34.74
7	ASKEx	32.28	64.41	49.31

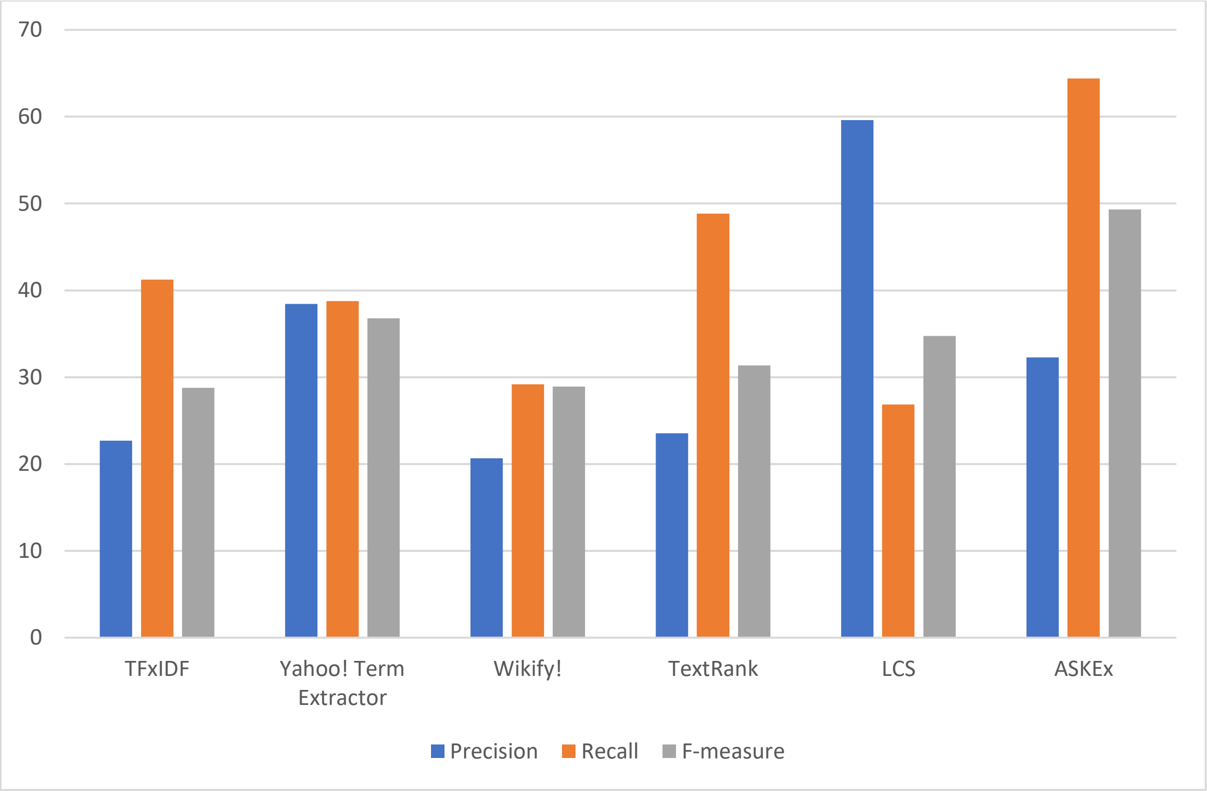


Fig 2: Bar graph of Performance evaluation extended by ASKEx modules

6. Future scope:

Utilising Natural Language Processing (NLP), term extraction has a bright future. There is an increasing demand for automated methods to extract usable information from unstructured text data as the amount of digital content keeps expanding exponentially [11]. One such technique is term extraction, which may be used to extract important ideas and subjects from text data and then use them in a variety of applications [17], including analysis, search, and recommendation. Future term extraction uses and developments using NLP include:

Multilingual term extraction:

Multilingual term extraction is becoming more and more necessary as organisations and digital material become more globally diversified. Multiple language texts can have their terms extracted using NLP approaches [13], which can then be applied to a variety of tasks like cross-lingual search and language translation.

Contextual term extraction:

NLP techniques can be used to extract terms in context, taking into account the surrounding words and phrases [5]. This can help improve the accuracy of term extraction and enable the extraction of more nuanced and specific concepts.

Deep learning-based term extraction:

Deep learning techniques such as neural networks can be used to extract terms from text data. These methods can learn to recognize patterns and relationships in text data, enabling more accurate and efficient term extraction.

Ontology-based term extraction:

Ontologies can be used to guide term extraction, enabling the extraction of more domain-specific concepts and improving the accuracy of term extraction.

Domain-specific term extraction:

NLP techniques can be customized for specific domains such as healthcare, finance, and legal, enabling the extraction of domain-specific terms and concepts [12].

Overall, term extraction using NLP has a promising future, with a variety of applications and advancements on the horizon. In natural language processing, there are several popular term extraction techniques, each with advantages and disadvantages. However, in the future, the most effective method for term extraction will probably depend on the particular context and use of the NLP system. A hybrid strategy that combines statistical, linguistic, and machine learning-based methods will probably be the best term extraction technique in the future to produce the best outcomes. In addition, context and domain-specific knowledge will become more crucial for correctly extracting pertinent terms from text.

7. Conclusion:

NLP techniques can be effective in automatically identifying and extracting relevant terms from a corpus of text. The accuracy of term extraction can be improved by using appropriate pre-processing techniques, such as stemming and stop-word removal.

The choice of NLP algorithm can have a huge impact on the quality of term extraction results [25], and researchers should carefully evaluate different options before selecting an approach. Term extraction can be useful in a various application, such as information retrieval, text classification and knowledge discovery [12].

Further research is needed to explore the potential of NLP techniques for more complex tasks, such as identifying multi-word expressions and disambiguating terms with multiple meanings.

In conclusion, ATR is an important task in NLP that can be useful in various applications. There are several methods available for ATR, each with its own strengths and weaknesses [10]. The choice of method depends on several factors like the size of the text collection, the domain of interest, and the availability of domain-specific knowledge resources. Further research is needed to develop more accurate and efficient methods for ATR in domain-specific text collections.

References:

[1] aclanthology.org

[2] link.springer.com

[3] Nisha Ingrid Simon, Vlado Kešelj. "Automatic Term Extraction in Technical Domain using Part-of-Speech and Common-Word Features", Proceedings of the ACM Symposium on Document Engineering 2018 - DocEng '18, 2018

[4] "PRICAI 2019: Trends in Artificial Intelligence", Springer Science and
Business Media LLC, 2019

[5] repository.smuc.edu.et (internet source)

[6] Using Wikipedia concepts and frequency in language to extract key terms
from support documents M. Romero ⇑ , A. Moreo, J.L. Castro, J.M. Zurita

[7] Srikanth Parameswaran, Srikanth Venkatesan, Manish Gupta. "Cloud Computing Security Announcements: Assessment of Investors' Reaction", Journal of Information Privacy and Security, 2014

[8] Unsupervised Technical Domain Terms Extraction using Term Extractor

[9] Automatic Term Extraction in Technical Domain using Part-of-Speech and Common-Word Features

[10] Conundrums in Unsupervised Key phrase Extraction: Making Sense of the State-of-the-Art

[11] Hulth A. ―Improved automatic keyword extraction given more linguistic knowledge‖, Proceedings of the 2003 conference on Empirical methods in natural language processing, pp. 216-223. Association for Computational Linguistics, Morristown, NJ, USA, 2003

[12] Caires, L.. "Elimination of quantifiers and undecidability in spatial logics for concurrency", Theoretical Computer Science, 20060807

[13] PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents

[14] Jiajia Feng et al., ―Keyword extraction based on sequential pattern mining‖, Proceedings of the Third International Conference on Internet Multimedia Computing and Service, pages 34-38, 2011

[15] Steier A., Belew R., ―Exporting phrases: A statistical analysis of topical language‖, Second Symposium on Document Analysis and Information Retrieval, 1993

[16] Patrícia Correia Saraiva, João M. B. Cavalcanti, Marcos A.Gonçalves, Katia C. Lage dos Santos et al. "Evaluation of parameters for combining multiple textual sources of evidence for Web image retrieval using genetic programming", Journal of the Brazilian Computer Society, 2012

[17] Qiankun Zhao, Sourav S. Bhowmick, Xin Zheng, Kai Yi. "Characterizing and predicting community members from evolutionary and heterogeneous networks", Proceedings of the 17th ACM conference on Information and knowledge management, 2008

[18] Ali Mehri et al., ―Keyword extraction by non-extensivity measure‖, Physical Review E, Volume 83, Issue 5, 2011

[19] M. Romero, A. Moreo, J.L. Castro, J.M. Zurita. "Using Wikipedia concepts and frequency in language to extract key terms from support documents", Expert Systems with Applications, 2012

[20] Shikha Mundra, Ankit Mundra, Anshul Saigal, Punit Gupta, Josh Agarwal, Mayank Kumar Goyal. "Chapter 56 Machine Learning Approaches for the Classification of Spammed Text in Messages", Springer Science and Business Media LLC, 2022

[21] scholars.ln.edu.hk

[22] Muhammad Yahya Saeed, Muhammad Awais, Muhammad Younas, Muhammad Arif Shah, Atif Khan, M. Irfan Uddin, Marwan Mahmoud. "An Abstractive Summarization Technique with Variable Length Keywords as per Document Diversity", Computers, Materials & Continua, 2021

[23] Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv (CSUR) 34(1):1–47

[24] Song L, Smola A, Gretton A, Bedo J, Borgwardt K (2012) Feature selection via dependence maximization. J Mach Learn Res 13(May):1393–1434

[25] Zhai C, Lafferty J (2004) A study of smoothing methods for language models applied to information retrieval. ACM Trans Inf Syst 22:179–214

[26] Methods for Automatic Term Recognition in Domain-Specific Text Collections: A Survey