Keyphrase extraction: A Survey

Khushboo Aggarwal[1], Vishal Gupta[2], Rohit Kumar[3]
UIET, Panjab University Chandigarh, India
(E-mail: a.khushboo1494@gmail.com[1], vishal@pu.ac.in[2], rklachotra@gmail.com[3])

Abstract— In this paper, survey of recent literature work for Keyphrase extraction is represented. Automatic keyphrase extraction is used to extract a small set of keywords or keyphrase from a given text document that can describe the meaning of whole document. It plays an important role in information retrieval as vast amount of information is present on web and summarizing it would reduce a lot of efforts of users. Thus reviewing latest literature work would help a lot in conducting further research.

Keywords—*Keyword extraction; Keyphrase extraction; Keyword; Keyphrase; Keyphrase extraction techniques(key words)*

I. INTRODUCTION

There is large amount of data present in most of the domain, i.e. News, education, social media, banking etc. All documents consist of so many sentences that divert us from main content or information provided by the document and takes too much time to analyze that what is discussed in document, so there is a need to define the terms that most describe the document so that it can be used for various purpose like in text summarization etc .

Automatic Keyword Extraction [7][22] is used for extracting Key word or phrases from text document to best describe the main content of document without any human intervention.

Limitation of manual extraction stating that it is time consuming and tedious task, an expensive or costly process as human resource is needed and no. of digitally available docs is constantly increasing. **Keyword:** [3] describes keyword as “A word that accurately and succinctly describes the context of document”. **Keyphrase** is also a phrase that accurately and succinctly describes the context of document. The main **difference** between Keyword and Keyphrase is that prior contains only a single word and later contains a group of words to describe the content. Intrinsic properties of Keyphrase are i) topic coverage that is percentage of content covered by keyphrase. ii) importance of topic covered by keyphrase. iii) Phraseness is the amount by which keyphrase is considered to be keyphrase in context of input language. iv) Informativeness mean coverage of main idea behind the text. Keyphrase extraction is useful in many fields like Education, Biomedical, Research and many more where data is present in the form of text.

A. General steps in Kephrase extraction [14]

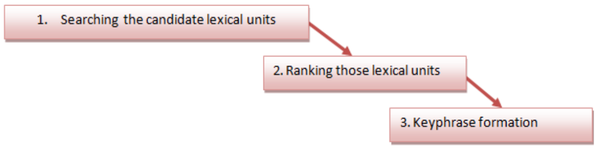
1. Searching the candidate lexical units
2. Ranking those lexical units
3. Keyphrase formation

Figure 1:Steps in keyphrase extraction.

In Figure1,First step is done by following some heuristic, for e.g. stop word removal or selecting words that are noun or adjective.

Second step includes measuring lexical unit’s importance through co occurrence characteristics or syntactic rules.

Third step is completed using top ranked lexical units.

B. Application of Kephrase extraction:

- Text highlighting: Extracted keyphrase of a document can be used to highlight keyphrase in document, this will enable readers to read the whole document really fast and also gets the idea of document in one go.

- Text summarization: Key terms provide subset of content which points out the main theme, concept or short idea of document used in summary generation. This helps in document evaluation that it is worth reading or not by readers[2].

- Information search: Key terms is very useful in searching on the web, with keyphrases, user could get better results by search engine in less time.

- Text categorization: Key terms are used to classify the text categorically that document belongs, for e.g. If text has word computer or information technology then the document belongs to IT category.

- Text clustering: Clustering strategies can be utilized to automatically group the fetched document into a rundown of significant classes on the basis of keyphrase extracted.

- Automatic Indexing: Indexing is where the server slithers through the site, gets each page that it can discover and stores a rundown of keyphrases that are found on the site in a database which is used to discover pages on your site when a user perform seek activities.

- Ontology learning: Ontology represents the knowledge within the domain . To make that model, Keyphrases are used where keyphrase extraction plays important role.

II. KEYPHRASE EXTRACTION APPROACHES

Figure 2 shows classification of Automatic keyphrase extraction whose points are discussed below.

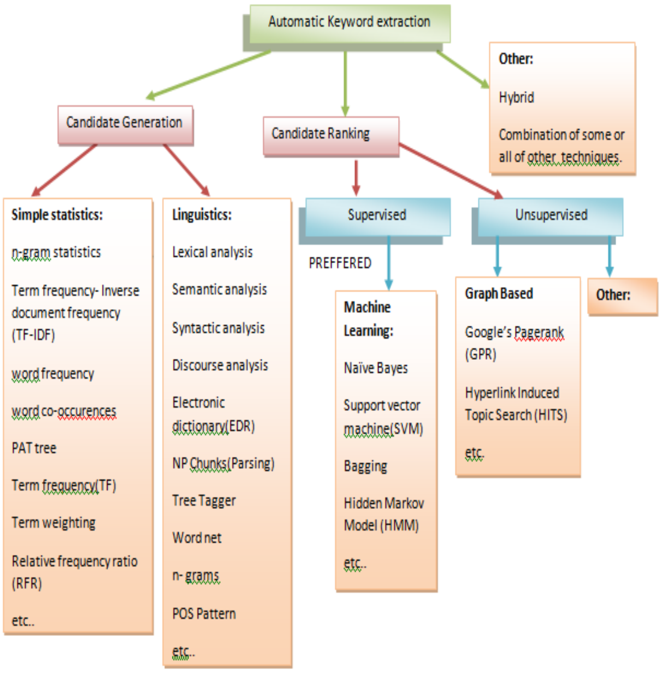

Figure 2:.Keyphrase extraction approaches.

Machine Learning: Machine learns from data provided to them for training. This training data can be supervised or unsupervised.[1]

In supervised, results of training data is given but not in case of unsupervised and machine is trained on basis of heuristic rules like in case of clustering.

For Keyword extraction, supervised techniques are more preferred.

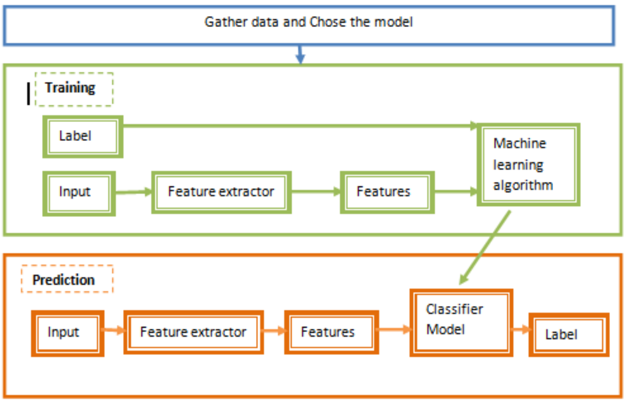

Figure 3:Machine Learning approach[8].

Figure 3 describes the machine learning approach where at 1st we gather annotated data and chose the model like SVM, HMM etc. Then model is trained. Here label represents the keyphrases extracted.

Simple statistics:[9] This type of strategy requires no training set . Their main focus is on statistics based on non linguistic feature of the document like word's position in document, the frequency of document etc as mentioned in figure 2.

Advantage: It is not dependent on any language or domain. It also require less processor and memory capacity.

Disadvantage: It is considered to be unrefined[4] in case of term frequency.

Linguistic: Under this approach linguistics feature of keywords are used for detection and extraction of keywords from document containing text.

Advantage: These approaches provide accuracy

Disadvantage: These are computationally expensive and also require domain knowledge with expertise in linguistics.

Graph based: Graph is considered to be mathematical model as it represents elements of text by use of nodes and edges represent the connection between related elements and thus providing us medium to explore the said relationship and structural information efficiently. Elements could be understood in terms of words, sentences etc.[5][6]

Hybrid:

The main motive of keyword extraction is to extract best keywords or phrases describing the document, thus hybrid of other mentioned techniques are used to retrieve the best results.

III. RECENT PAPER

A. KeyRank

Wang, Sheng and Wu (2018) [10] proposed a new idea for extracting Keyphrase named KeyRank(Unsupervised approach) containing two components KCSP (searching candidate) and PF-H(ranking candidate) as in Figure 4 for extracting keyphrase from specific document and performed experiment on 2 dataset SemEval-2010 containing 244 articles and INSPEC containing 2000 abstracts.

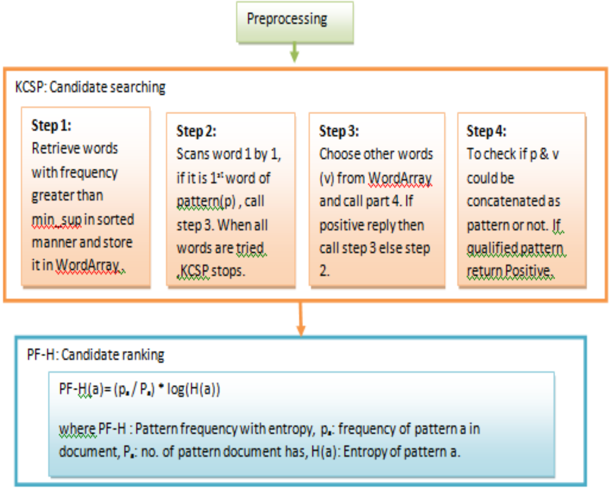
Preprocessing
KCSP: Candidate searching
Step 1: Retrieve words with frequency greater than min_sup in sorted manner and store it in WordArray.
Step 2: Scans word 1 by 1, if it is 1st word of pattern(p) , call step 3. When all words are tried ,KCSP stops.
Step 3: Choose other words (v) from WordArray and call part 4. If positive reply then call step 3 else step 2.
Step 4: To check if p & v could be concatenated as pattern or not. If qualified pattern return Positive.
PF-H: Candidate ranking
PF-H(a)= (pa / Pa) * log(H(a))
where PF-H : Pattern frequency with entropy, pa: frequency of pattern a in document, Pa: no. of pattern document has, H(a): Entropy of pattern a.

Figure 4: KeyRank

KCSP is algorithm for candidate search from specific document using pattern mining in sequence with gap constraints, having benefits i) document is scanned only once which ii) reduce the time consumption for setup and work done by human, and iii) gap constraint is treated as inside property and iv) improves the exactness and relevance and reducing computation time(i.e. provides compact results). Its disadvantage is that it ignores the inside property of usefulness of pattern, duplication removal takes more time.

PF-H(Pattern frequency with entropy) is a mechanism used for key phrase candidate ranking using PF-IDF(pattern frequency-inverse document frequency) and depends on 3 probabilities i) independent form ii) sub form iii) of other situations. It justifies meaningfulness, usefulness, and accuracy of pattern. Experiment results shows that KeyRank performance is better than KeyEx, TextRank, TextRank-A in context of precsion, recall, F-score for Sem eval data(documents) and precision and Fscore becomes less than TextRank-A for less no. of keyphrase extracted for INSPEC(abstract data).

Future work : can be done for improving it for less no. of keyphrase and also when words do not repeat in keyphrase.

B. *SwiftRank*

Lynn, Lee and Kim(2017)[11] proposed the unsupervised statistical technique named SwiftRank for Keyword and salient sentence extraction which considers that terms related to corresponding title contains salient information and its importance depends on position of sentence in text.

For this, sentence score is calculated such that sentences at beginning and end have more importance than sentence at mid position.

And Term score is also normalized so that score of unigrams and shorter phrase dominates the ranking and assumes that longer phrase could lead to insignificant information. The candidate keyword or keyphrase is extracted when term score > mean value of normalized term score.

$$T_{s_t,t_k} = \sum_{k=1}^{n} f(t_k) + S(s_t)$$

where T t_k , s_t : Unigram score for encountered word in sentence, $f(t_k)$: Term frequency of kth word, $S(s_t)$: sentence score

$$S(s_t) = (|P(s_t)'| * \frac{l(s_t)}{\max Ls}) + (Tt(s_t) * \frac{t(s_t)}{\max Ws})$$

where $l(s_t)$: *t*th line s's length in document, max L_s : maximum length of line in set of sentences , $Tt(s_t)$: quantity of heading terms encountered in *t*th line *s* , $t(s_t)$: the quatity of words in *t*th line after removal of stopwords, max W_s: maximum amount of words in single line among set of sentences, | $P(s_t)$'| : line position score.

$|P(s_t)'| = |P(s_t) - (N/2)|$

where s_t belongs to set of sentences, N : Total no. of sentences.

For experiment dataset used was set of 600 news article and results are compared with TextRank and RAKE which shows that for keyword extraction it outperforms both but for larger window size, its performance decreases to itself, and for sentence extraction it outperforms TextRank for Precision and F-score but lags behind for Recall.

Benefits of the described model are 1)some work of pre-processing is removed such as finding name entities, noun chunking, and POS tagging which really decreases the setup time. 2) It is not language dependent. 3) Its accuracy is high and proves to be efficient.

Future work: More preprocessing simplification could be done and this approach could be combined with other algorithms for better and logical results.

C. *SPMW,KeyEx*

Xie, Wu and Zhu (2017)[12] proposed an algorithm for efficient sequential pattern mining with wildcards for keyphrase extraction[Figure 5]. The main point discussed by them is the use of wildcards for extracting sequential patterns so that gap constraint within the pattern could establish the semantic relationship between words as pattern diversity and flexibility creates problem in it and existing methods can't find document specific patterns. It uses a supervised learning approach.

This paper consist of two parts: 1) Efficient sequential pattern mining with wildcard(SPMW) and one-off conditions: For the discovery of sequential patterns from sequences, combination of both are used for deriving pattern. Wildcards are used to find out the logical meanings from text, one-off condition is used to capture important keywords during pattern driven process. (2) Keyphrase extracted from specific document(KeyEx): Here combination of patterns found from each document data and machine learning approach is used to extract keyphrases. Such a pattern based strategy has two points of interest: (a) the sequential pattern are explicit for every document and are autonomous of the whole dataset (b) Keyphrase learning procedure can adjust to records from various areas.

KeyEx uses baseline and pattern features. Wildcards are categorized in three types : (1) continuous patterns in sequence (without wildcards), (2) variable gap sizes (with minimum and maximum support threshold values specified by user), and (3) large sizes. Then according to mined patterns and statistical information, using naïve bayes algorithm, model is trained and probability value is assigned to each phrase so that top k keyphrase is selected among all.

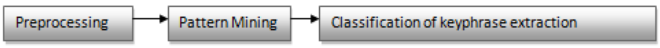

Figure 5: steps in algorithm.

For experiment Datasets used was Reuters-21578 collection and SemEval 2010. Results show that when no. of selected keyphrase varies from 3 upto 25, F1 score first increase then decrease, best F1 value is obtained when the support threshold is set to 5. KeyEx performs better than Kea.

Experiment Results shows that best results are obtained when maximum gap size are set to 1 and 2.

Future work: This algorithm is with supervised approach and could be extended for unsupervised one also.

D. TSAKE

Asla and Nickabadib et al.[13] proposed an algorithm for topical and structural automatic keyphrase extractor. It is a graph based approach and it opt for the topic model for weighing edges instead of cooccurrence graph's nodes.

TSAKE applies network analysis technique to each topical graph considering the 1)topic coverage 2) its importance 3) Phraseness 4) Informativeness of Keyphrase

At preprocessing, Topic Model is built using Wikipedia documents and structural, statistical and semantic information is used to form cooccurrence graph(complexity O(V^2)) then candidate phrases are scored in different topics and on its basis top N is selected, then community detection algorithm is used to find minor topic, then centrality measures are applied to obtain central nodes and its associated words are used to score candidate Keyphrase.

Dataset used for this is SemEval 2010 and Hulth containing abstracts and Marujo containing news stories.

Results shows that centrality approach provides best results, LDA based method provides comparable results.

TSAKE extracts high quality and meaningful keyphrases even where other baseline methods fails.

Future work: Macro and micro topics could be combined into one procedure and instead of topical N ram other models could also be used for better performance.

E. PKEA

Jie Hu et al.(2018)[14] proposed an algorithm named patent keyword extraction algorithm (PKEA) based on distributed representation and skip gram model, k-means algo, cosine similarity are used for patent classification.

Skip gram:- It is an algorithm based on deep learning neural net to train that encodes word into real valued, dense and low dimensional vectors which represent semantic and syntactic relation between words.

K mean algo- used to find out centroid vector from the vectors obtained from skip gram model. Basically, given data is partitioned ino k clusters.

Cosine similarity- used to obtain similarity value list sorted from largest to smallest values with centroid word and top n words are selected for each document.

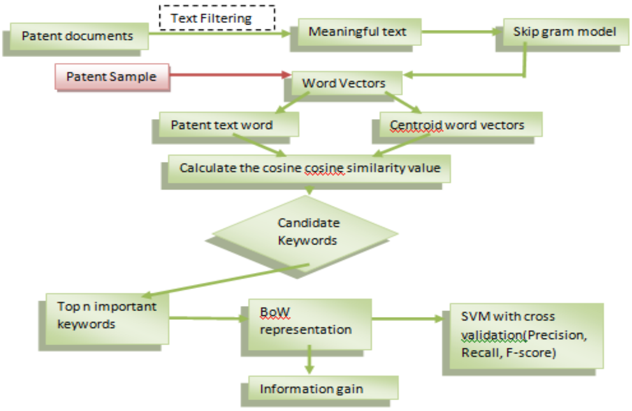

Figure 6:- Overall process

PKEA:- Extract keyword from patent text.

Evaluation is done using two methods.1) microcosmic- Information gain theory is used to measure importance of each extracted keyword. Higher the IG score better the performance is.2) SVM- classification is done by using SVM(support vector machine) with linear kernel. Then precision, recall, F1 score are used for evaluation.

Datasets considered are 1) autonomous cars patent corpus containing five (5*500 documents) different categories of patent.2) SemEval 2010 containing ACM documents.

When compared with Tf- IDF, RAKE, TextRank , PKEA performed really well for extracting small set of keywords.

Future work:- addition of position features to train word embedding for more meaningful keyword extraction.

F. Key-LUG

N.Giamblanco and P. Siddavaatam (2017) [16] proposed Keyword extraction system named Key-LUG which is unsupervised approach for single document and is domain independent. Authors proposed that Approach proposed by Newton for gravitational theory could be applied to linguistics through which word could be clustered. For this, words would be considered as physical quantity of mass and Newton law defines the interaction between those masses.

The whole model is divided into four steps: (i) Noise filtering:- Similar as preprocessing. (ii) Word mass assignment:- Importance of word is defined by its frequency in document and character length as longer the word, more relevant it is. Thus, it is product of these two. Relative location of each word is considered as distance between them is considered. (iii) Word attraction computation:-Proposed model is network of words which could be assumed as complete weighted graph. $F_{t_{ij}} = G \frac{m_{t_i} m_{t_j}}{|r_{ij}|^2}$

This equation is used for computation where Document contains n words $d = \{t_{0,}t_{1,}.....,t_n\}$ and t_i occurs at different positions $t_i = \{p_{0,}p_{1,}....,p_w\}$. here w defines relationship between t_i and t_w . G is constant and m_{t_i}, m_{t_j} are word masses.

(iv) Keyword and keyphrase ranking:- After computation, those computed values are sorted with respect to F forming a set S from which K subset of keyword pair is selected which has maximum force.

Dataset considered for experiment is SemEval 2010.

Its benefit is that it completely captures the meaningful text but its limitation is that it is only applicable for bi word keyphrase extraction. But for biword, results are very good.

G. RVA

E. Papagiannopoulou and G. Tsoumakas(2018) [17] proposed a local word vectors based keyphrase extraction technique which is unsupervised and used GloVe technique which is used to generate word vectors corresponding to local word embedding.

The fundamental idea here is to provide neighborhood of each word and its local contexts. The reference vector is calculated by averaging the local word vectors of the title and abstract of a document affected by word occurrence. Later, cosine similarity has been calculated among unigram's local vector and reference vector for ranking purpose.

The RVA (Reference vector algorithm) is used to implement the concept having 2 steps: (i)Candidate keyphrase production:-candidate keyphrases is restricted upto trigram.(ii) Scoring the candidate keyphrase:- firstly, local word vectors are computed using GLoVe technique which picturise local context then reference vector is computed after which cosine similarity is computed for each candidate term w.r.t. full and reference text.

Dataset considered are Krapivin 2008 and SemEval 2010 and evaluated on basis of F1 score.

Its benefit is that some preorcessing steps is not required which reduces the time consumption but it is only limited upto trigram keyphrases and also this technique is useful when keywords are to be extracted from abstract and title and not from full text.

Future work:-Graph based unsupervised and supervised methods could be employed using local word embeddings for full text which does not get affected by noise and redundancy contained in document.

H. EmbedRank

K.B. Smires, C. Musat, A Hossmann, M. Bareriswyl, M. Jaggi (2018).[18] proposed Embed Rank which is unsupervised, corpus independent approach based on phrase and document embeddings. MMR(Goldstein,1998) is used for diversifying result due to its simplicity in implementation and interpretation both. For ensuring informativeness, meaningful distance between candidate phrase and document is calculated and for diversity, semantic distance between candidates is calculated. It uses Sent2Vec (Pagliardini et al., 2017) for producing sentence embeddings which provides semantic relatedness between phrases.

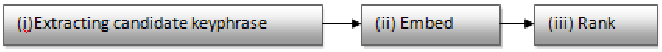

Figure 7:Steps for EmbedRank

First step is done by using POS sequences keeping sentences with which keep words containing adjective followed by nouns. Second step covers calculation of document(doc2Vec, Sent2Vec) as well as each candidate keyphrase embeddings ,Then cosine similarity is used between candidate phrase and document embedding. Third step includes ranking of candidate phrases w.r.t. their cosine distance from document embedding.

Now problem of redundancy occurs which is resolved by using EmbedRank++ which uses MMR(Maximal Marginal relevance) which combines the relevance and diversity.

$$MMR := \overset{argmax}{A_i \epsilon R/S} [\gamma . Sim_1(A_i, q) - (1 - \gamma) \overset{max}{A_i \epsilon S} Sim_1(A_i, A_j)]$$

here, R is set of retrieved document, q is input query, S is set of documents that are good for query, A_i, A_j are retrieved documents and Sim is similarity function.

Dataset considered are Inspec(Hulth 2003) consisting short docs, DUC 2001(Wan and Xiao,2008) containing medium length docs and NUS(Nguyen and kan,2007) containing long docs.

It concludes that Sent2Vec is better approach then Doc2Vec and also raises question of F score as an evaluation measure based on their study conducted.

Its benefit is that extraction process is fast and keyphrases extracted are disjoint making them highly readable and ensures informativeness and diversity, it is corpus free approach, enables real time computation, Grouping avoids overgeneration problem. Its application is mainly for information retrieval task

I. Graph and cluster

Y. Ying and L. Panpan(2017) [19] proposed the graph and cluster method where it considers connection between sentence and words as if sentence is important then word appearing in it would also be important and adopts topic clustering algo for it. It uses unsupervised graph base approach for determining correct keyphrase.

In proposed model, at first stop words and identical words are removed, then secondly, 3 types of graphs are formed:- (i) Sentence-to- Sentence graph(G_{ss} Graph): Representing sentence as vector and then finding similarity between 2 sentences using cosine. Then graph is formed containing sentences as nodes and similarity between nodes as edges (ii) Word-to-Word graph(G_{ww} Graph) : Word embedding is used to find similarity of 2 words using SENNA as training method, then semantic relatedness is found out using cosine similarity on basis of which, graph is made. (iii) Sentence to word graph(G_{sw} –Graph):This undirected graph is made based on G_{ss} and G_{ww}. Edge exist only when word is present in sentence. After making these graphs Ranking is done on basis of 2 assumption i.e. word is important if it is connected to other important word or it appears in important sentence. For keyphrase extraction, term clustering is applied on word graph. Kmeans clustering is used for the purpose.

Dataset used are Hulth 2003 and Luis Marujo called 500N

Its main benefit is that it covers all major topic of document but its limitation is that it corresponds to only single document.

Future work could be done by modifying clustering method and also considering whole corpus instead of single document.

J. RankUp

Gerardo Figueroa ,P. Chen and Y. Chen (2018) [20] proposed an unsupervised graph based keyphrase extraction with error feedback algorithm named Rankup which applies error feedback mechanism using backpropagation to enhance the graph-based(TextRank and RAKE) unsupervised alogorithms..

The method has five main stages: (i)Graph construction: Used for text where text unit is node and relation between them is edges and this is accomplished by using TextRank or RAKE (ii) Node ranking: On the basis of graph structure nodes are ranked via ranking algorithm. TextRank uses recursive algorithm and RAKE calculate it in single pass. (iii)Error Detection: Each node is evaluated to a permissible rank using error detection approach, for this 3 approaches are used named TFIDF, RIDF, Clusteredness.(iv) Error feedback: Errors are back-propagated to adjust the weights of edges goals to minimize the difference between current node score and expected node scores. (v)Keyphrase output: The resulting keyword are outputted by sorting them in descending order according to scores and .then top n keyphrases are selected to output according to need.

Datasets used are Kaggle containing 1876 user question on diverse topic, Hulth 2003 containing 2000 abstracts, IEEE Xplore for which a web crawlers made to crawl the document and extract 417 abstracts belonging to scientific paper.

The main point in this paper is merging of unsupervised technique with supervised concept of error feedback.

Future work includes performing experiment with different algorithms like graph Construction, node ranking etc. different measures could be tested in error detection stages. Secondly, overcoming upper bound limit on performance due to missing of keyphrase in actual text.

K. CopyRNN

Meng et al. (2018)[21] proposed CopyRNN technique which is supervised approach for keyphrase prediction with encoder decoder framework. To capture both semantic and syntactic features, RNN(Recurrent neural network) is used.

Three steps are included in this model:- (i) Source text and multiple target phrase sequences are converted into text-keyphrase pairs containing one source sequence and one target sequence corresponding to it.(ii) Now encoder (GRU(gated recurrent unit)) and decoder (forward GRU) model is applied to be trained with mapping from source to target sequence. while encoding, RNN convert variable length input sequence to set of hidden representation from which context vector is obtained. Then at decoding time, context vector is decoded into variable length sequence through conditional language model .(iii) Copy Mechanism is applied to predict out of vocabulary words by selecting appropriate word from the input text.

Dataset used are Inspec(Hulth 2003) containing 2000 abstracts, Krapivin(Krapivi et al., 2008) containing 2304 papers with full text and author assigned keyphrases, NUS(Nguyen and Kan, 2007) containing 211 papers, SemEval(Kim et al., 2010) containing 288 articles from ACM digital library, KP20k built by authors of this paper containing title, abstract, keyphrases of 20000 scientific articles.

Its main benefit is to find out absent keyphrases and hidden semantics behind the text from the source text and keyphrases could be extracted from unfamiliar text but limitation is that it gives priority to shorter keyphrases and some phrase appears to be semantically similar to keyphrase is given as output.

Future work: The location of core information could be extended to other formats like image, videos.

IV. RESULTS AND COMPARISONS

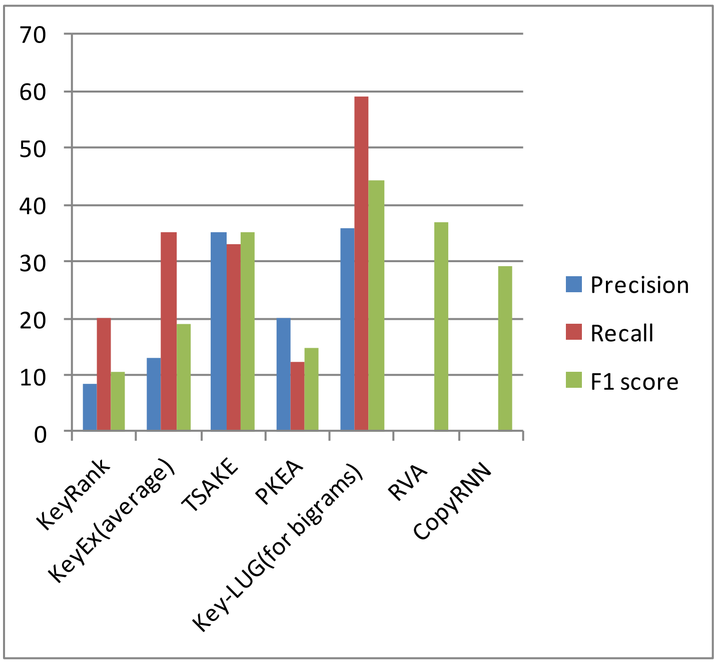

Figure 8: Comparison of different technique on SemEval 2010 dataset

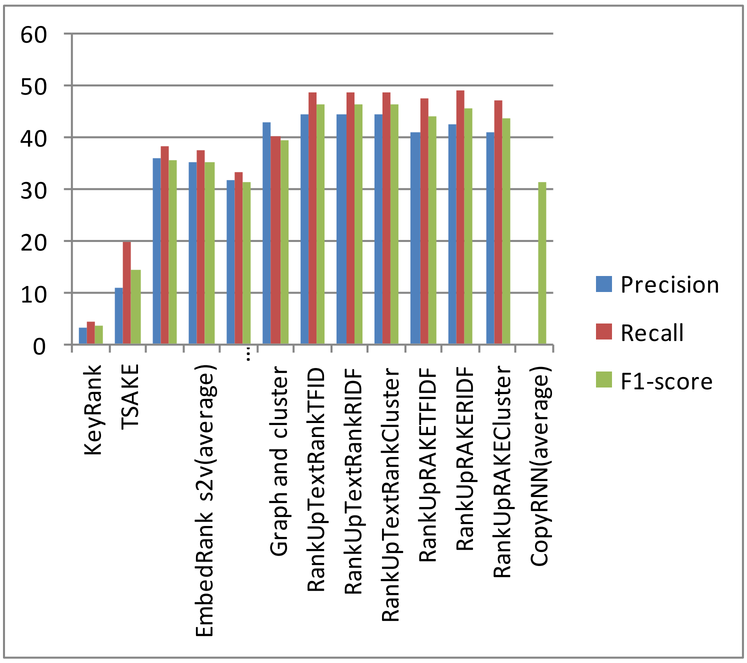

Figure 9: Comparison of different technique on Inspec 2003 dataset

TABLE II : Results contained in discussed papers

Methods/ Contributor	Approach	Dataset	Evaluation Measure	Results (P=Precision, R=Recall, F=F-score, WS=Window size MST=minimum support threshold)

KeyRank (Wang et al(2018)) [10]	Simple Statistics, Linguistics, Unsupervised	SemEval-2010 containing 244 articles	Precision, Recall, F-score	If MST =3:
				P=8.33%
				R=19.94%
				F1=10.49%

		INSPEC containing 2000 abstracts	-do-	If MST =3:
				P=3.12%
				R=4.32%
				F1=3.39%

SwiftRank(Lee et al(2017)) [11]	Simple Statistics, Unsupervised	web document collection containing 600 news articles (several media sources) (300 each	For Keyword extraction- Precision, recall, F-score	For WS =5: (P=0.784, R=0.793, F=0.788) For WS =10: (P=0.585, R=0.592, F=0.588) For WS =15: (P=0.503,

		for validation and testing)		R=0.498, F=0.5)
			For sentence Extraction- Precision, recall, F-score under Rouge-N where N represent no. of grams	ROUGE-I: (P=0.518, R=0.490, F=0.498) ROUGE-II: (P=0.540, R=0.492, F=0.509) ROUGE-III: (P=0.544, R=0.476, F=0.503)
SPMW, KeyEx (Xie et al (2017))[12]	Linguistics, Machine learning, Supervised	Reuters-21578 collection and SemEval-2010	For SPMW: number of patterns	If MST = 30: 176873 If MST = 35: 46441 If MST = 40: 2752
			For KeyEx: Precision, Recall, F1-score	If MST=2 (average): (P=0.130, R=0.3544, F=0.1901) If MST =3 (average): (P=0.1301, R=0.34885, F=0.1895)
TSAKE (Asla et al, (2017))[13]	Graph based, Machine learning, Supervised	SemEval-2010(144 for training,40 for validation and 100 for testing among 244 articles)	Precision, Recall, F-score	SemEval-R: (P=26.6%, R=33.8%, F=29.8%) SemEval-A: (P=14.3%, R=56.8%, F=22.8%) SemEval-C: (P=35.1%, R=33.2%, F=35.1%)
		Hulth(2000 abstracts)	-do-	Hulth-U: (P=40.1%, R=20.3%, F=26.9%) Hulth-C: (P=11.1%, R=19.9%, F=14.3%)

		Marujo(500 news stories)	-do-	P=14.3%
				R=46.6%
				F=21.9%

PKEA(Jie Hu et al. (2018) [15]	Simple Statistics, Linguistics, Machine Learning(Neural network) Supervised	autonomous cars patent corpus(5 categories with 500 documents in each category of patent)	Precision, Recall, F1-score	No. of keywords=~24 :
				P=81.61%(highest)
				R=82.76%(highest)
				F1=82.31%(highest)

		SemEval-2010(144 for training,100 for testing)	-do-	SemEval-R: (Average) (P=16.33%, R=12.33%, F1=13.5%) SemEval-C: (Average) (P=20.03%, R=12.37%, F1=14.73%)

Key-LUG (N.Giamblanco et al.) (2017)[16]	Simple statistics, Linguistics, Unsupervised	SemEval-2010 containing 244 articles	Precision, Recall, F1-score	P=35.71%
				R=58.92%
				F1=44.42%

RVA (E.Papagiannopoulou et al)(2018)[17]	Unsupervised	Krapivin(2304 scientific text articles)	F1-score	F1=0.32062
		SemEval-2010	F1-score	F1=0.36815
(i) EmbedRank d2v (Bennani-Smires et al.)(2018)[18]	Lnguistics, Unsupervised	Hulth 2003(2000 journal abstracts)	Precision, Recall, Macro F1-score	For WS =5: (P=41.49, R=25.40, F1=31.51) For WS =10: (P=35.75, R=40.40, F1=37.94) For WS =15: (P=31.06, R=48.80, F=37.96)
		DUC 2008(308 newspaper article)	-do-	For WS =5: (P=80.87, R=19.66, F1=24.02) For WS =10: (P=25.38, R=31.53, F1=28.12)

				For WS =15: (P=22.37, R=40.48, F1=28.82)
		NUS 2007(211 long documents)	-do-	For WS =5: (P=3.88, R=1.68, F1=2.35) For WS =10: (P=3.95, R=3.28, F1=3.58) For WS =15: (P=4.33, R=5.89, F1=4.99)
(ii) EmbedRank s2v		Hulth 2003(2000 journal abstracts)	-do-	For WS =5: (P=39.63, R=23.98, F1=29.88) For WS =10: (P=34.97, R=39.49, F1=37.09) For WS =15: (P=31.48, R=49.23, F1=38.40)
		DUC 2008(308 newspaper article)	-do-	For WS =5: (P=34.84, R=22.26, F1=27.16) For WS =10: (P=28.82, R=35.58, F1=31.85) For WS =15: (P=24.49, R=44.20, F1=31.52)
		NUS 2007(211 long documents)	-do-	For WS =5: (P=5.53, R=2.44, F1=3.39) For WS =10: (P=5.69, R=5.18, F1=5.42) For WS =15: (P=5.34, R=7.06, F1=6.08)
(iii)EmbedRank++ s2v		Hulth 2003(2000 journal abstracts)	-do-	For WS =5: (P=37.44, R=22.28, F1=27.94) For WS =10:
				(P=30.31, R=34.29, F1=32.18) For WS =15: (P=27.24, R=43.25, F1=33.43)
		DUC 2008(308 newspaper article)	-do-	For WS =5: (P=24.75, R=16.20, F1=19.58) For WS =10: (P=18.27, R=23.34, F1=20.50) For WS =15: (P=14.86, R=27.64, F1=19.33)
		NUS 2007(211 long documents)	-do-	For WS =5: (P=2.78, R=1.24, F1=1.72) For WS =10: (P=1.91, R=1.69, F1=1.79) For WS =15: (P=1.59, R=2.06, F1=1.80)

Graph and cluster(Ying et al)(2017) [19]	Linguistics, graph based, unsupervised	Inspec 2003,	Precision, Recall, F-score	P=43%
				R=40.2%
				F=39.6%

		Marujo	-do-	P=48.7%
				R=49.8%
				F=47.8%

RankUp^{T} extRank$_{TFIDF}$(Figueroa et al.) (2018) [20]	Statistical, linguistic, graph based, Unsupervised	Kaggle	Precision, Recall, F1-score	P=25.4%
				R=29.9%
				F1=27.5%

		Inspec 2003	-do-	P=44.3%
				R=48.8%
				F1=46.4%

		IEEE Xplore Collection	-do-	P=24.5%
				R=27.0%
				F1=25.7%

RankUp^{T} extRank$_{RIDF}$		Kaggle	-do-	P=24.9%
				R=29.5%
				F1=27.0%

		Inspec 2003	-do-	P=44.4%
				R=48.9%
				F1=46.6%

		IEEE Xplore Collection	-do-	P=23.8%
				R=26.3%
				F1=25.0%

$\text{RankUp}^{\text{TextRank}}_{\text{Cluster}}$		Kaggle	-do-	P=24.0%
				R=28.8%
				F1=26.2%

		Inspec 2003	-do-	P=44.3%
				R=48.7%
				F1=46.4%

		IEEE Xplore Collection	-do-	P=22.3%
				R=24.9%
				F1=23.5%

$\text{RankUp}^{\text{RAKE}}_{\text{TFIDF}}$		Kaggle	-do-	P=32.7%
				R=37.5%
				F1=34.9%

		Inspec 2003	-do-	P=41.2%
				R=47.5%
				F1=44.1%

		IEEE Xplore Collection	-do-	P=31.9%
				R=34.9%
				F1=33.3%

$\text{RankUp}^{\text{RAKE}}_{\text{RIDF}}$		Kaggle	-do-	P=30.4%
				R=36.0%
				F1=33.0%

		Inspec 2003	-do-	P=42.7%
				R=49.3%
				F1=45.7%

		IEEE Xplore Collection	-do-	P=29.9%
				R=32.7%
				F1=31.2%

$\text{RankUp}^{\text{RAKE}}_{\text{Cluster}}$		Kaggle	-do-	P=28.8%
				R=33.6%
				F1=31.0%

		Inspec 2003	-do-	P=41.1%
				R=47.2%
				F1=43.9%

		IEEE Xplore Collection	-do-	P=27.1%
				R=30.4%
				F1=28.7%

CopyRNN (Meng et al) (2018)[21]	Linguistics Machine learning, Supervised	Inspec, NUS	F1 score	For WS =5:
				F1=0.292
				For WS =10:
				F1=0.336

		Krapivin	-do-	For WS =5:
				F1=0.302
				For WS =10:
				F1=0.252

		NUS	-do-	For WS =5:
				F1=0.342
				For WS =10:
				F1=0.317

		SemEval	-do-	For WS =5:
				F1=0.291
				For WS =10:
				F1=0.296

		KP20k	-do-	For WS =5:
				F1=0.328
				For WS =10:
				F1=0.255

V. CONCLUSION

Keyphrase extraction is very useful technique for many applications as discussed. And in recent years lots of work have been done in this field.

Supervised approaches need large amount of training data and its generalization outside domain is also poor and for unsupervised approach also accuracy is not good enough and generalization problem also persist.[18][20]

Graph based method has major drawback that frequently used term get higher score as number of edges get increased and rare terms get lower scores.[19][20].Longer the document, Positional information becomes more important and using them results are unbeatable by any other method[17].

According to observed values on given surveyed data, TSAKE performs well on SemEval dataset compared to other mentioned techniques[figure] and $\text{RankUp}^{\text{TextRank}}_{\text{RIDF}}$ performs well on Inspec dataset compared to other mentioned techniques even better than TSAKE[figure]. Thus it can be concluded that RankUp is more better approach to be implemented.

RankUp can be combined with different algorithms for further improved results. All techniques have their own advantage and disadvantage but uptil now, results are not upto the mark against human annotators as For these evaluation measures, no value crosses above 60%.

[18] also raises question on F1 score as evaluation measure.

There is also a gap in all these document that Keyphrase extraction is dependent on language and an approach is needed which is language independent.

REFERENCES

[1] J. Brownlee, "A Gentle Introduction to Text Summarization", *Machine Learning Mastery*, 2018. [Online]. Available: https://machinelearningmastery.com/gentle-introduction-text-summarization/. [Accessed: 22- Dec- 2018]

[2] B. Santosh Kumar, B. Korra Sathya and J. Sanjay Kumar, "Automatic Keyword Extraction for Text Summarization: A Survey", 2017.

[3] S. Siddiqi and A. Sharan, "Keyword and Keyphrase Extraction Techniques: A Literature Review", *Research.ijcaonline.org*, 2015. [Online]. Available: https://research.ijcaonline.org/volume109/number2/pxc3900607.pdf. [Accessed: 01- Nov- 2018]

[4] S. K. Bharti and K. S. Babu, "Automatic Keyword Extraction for Text Summarization: A Survey," *Int. Conf. Informatics Anal. ACM*, p. 86, 2017.

[5] S. Beliga, "Keyword extraction: a review of methods and approaches," *Univ. Rijeka, Dep. Informatics, Rijeka*, pp. 1–9, 2014.

[6] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artif. Intell. Rev.*, vol. 47, no. 1, 2017.

[7] S. Siddiqi and A. Sharan, "Keyword and Keyphrase Extraction Techniques: A Literature Review," *Int. J. Comput. Appl.*, vol. 109, no. 2, pp. 18–23, 2015.

[8]"6. Learning to Classify Text", *Nltk.org*, 2018. [Online]. Available: https://www.nltk.org/book/ch06.html. [Accessed: 26- oct- 2018]

[9]2018. [Online]. Available: https://ieeexplore.ieee.org/document/5966451. [Accessed: 26- Oct- 2018]

[10]S. Siddiqi and A. Sharan, "Keyword and Keyphrase Extraction Techniques: A Literature Review", *Research.ijcaonline.org*, 2015. [Online]. Available: https://research.ijcaonline.org/volume109/number2/pxc3900607.pdf. [Accessed: 01- Nov- 2018]

[11] H. M. Lynn, E. Lee, C. Choi, and P. Kim, "SwiftRank: An Unsupervised Statistical Approach of Keyword and Salient Sentence Extraction for Individual Documents," in *Procedia Computer Science*, 2017, vol. 113, pp. 472–477.

[12] F. Xie, X. Wu, and X. Zhu, "Efficient sequential pattern mining with wildcards for keyphrase extraction," *Knowledge-Based Syst.*, vol. 115, pp. 27–39, 2017.

[13] Q. Wang, V. S. Sheng, and X. Wu, "Document-specific keyphrase candidate search and ranking," *Expert Syst. Appl.*, vol. 97, pp. 163–176, 2018.

[14] J. Rafiei-Asl and A. Nickabadi, "TSAKE: A topical and structural automatic keyphrase extractor," *Appl. Soft Comput. J.*, vol. 58, pp. 620–630, 2017.

[15] J. Hu, S. Li, Y. Yao, L. Yu, G. Yang, and J. Hu, "Patent Keyword Extraction Algorithm Based on Distributed Representation for Patent Classification," *Entropy*, vol. 20, no. 2, p. 104, 2018.

[16] N. Giamblanco and P. Siddavaatam, "Keyword and Keyphrase Extraction using Newton's Law of Universal Gravitation," in *Canadian Conference on Electrical and Computer Engineering*, 2017.

[17] E. Papagiannopoulou and G. Tsoumakas, "Local word vectors guiding keyphrase extraction," *Inf. Process. Manag.*, vol. 54, no. 6, pp. 888–902, 2018.

[18] K.Bennani-Smires, C.Musat, A.Hossmann, M.Baeriswyl and M.jaggi, "Simple Unsupervised Keyphrase Extraction using Sentence Embeddings", arvix.org, 2018. [Online]. Available: https://arxiv.org/abs/1801.04470.

[19] Y. Ying, T. Qingping, X. Qinzheng, Z. Ping, and L. Panpan, "A Graph-based Approach of Automatic Keyphrase Extraction," in *Procedia Computer Science*, 2017, vol. 107, pp. 248–255.

[20] G. Figueroa, P. C. Chen, and Y. S. Chen, "RankUp: Enhancing graph-based keyphrase extraction methods with error-feedback propagation," *Comput. Speech Lang.*, vol. 47, pp. 112–131, 2018.

[21] R.Meng, S. Zhao, D.He and P.Brusilovsky "Deep Keyphrase generation" *IEEE Access*, vol. 6, pp. 46047–46057, 2018.

[22] S. Kazi and N. Vincent, "Automatic Keyphrase Extraction: A Survey of the State of the Art", *Aclweb.org*, 2014.[Online].Available:http://www.aclweb.org/anthology/P14-1119. [Accessed: 08- Dec- 2018]