Automatic Extractive Text Summarization using K-Means Clustering

Krithi Shetty
Department of Computer Science and Engineering
M S Ramaiah Institute of Technology, Bangalore, India
krithishetty8@gmail.com

Jagadish S Kallimani
Department of Computer Science and Engineering
M S Ramaiah Institute of Technology, Bangalore, India
jagadish.k@msrit.edu

***Abstract*-- The rise in the dimension of the World Wide Web has made an explosion of the amount of accessible information. As the textual data involves several instances of redundancy, omission of part of sentences or entire sentences is possible without altering the meaning of the document. Summarization of the text can informally be defined as the act of condensing the document from its original size without significantly compromising the semantics. For the purpose of generating an appropriate summary, the raw text is first pre-processed which involves - removing non-ASCII characters and stop-words, tokenizing and stemming. Appropriate features are extracted from the data, *tf-idf* values for each word are computed and the entire pre-processed data is then transformed into a *tf-idf* matrix. Every sentence of the document will be represented as a vector in the dimensional space of the document's vocabulary. To obtain a concise summary, sentences are appropriately clustered based on the degree of separation of vectors in the Euclidean place. Association of sentences to a cluster using K-means method is totally based on cosine similarity. The count of the clusters is to be formed is predefined. As the number of clusters increase the accuracy of the summary increases. From each of the clusters the sentences which are informative are picked to form the final summary. Using *recall* and *precision* measures, the effectiveness of the summary is verified.**

Keywords-- ***Finite State Machine, Vector Space Model, Clustering, Lemmatization, Cosine Similarity, Sentence Extraction, Summary Generation.***

I. INTRODUCTION

Redundancy of words is recurrent in all natural languages. Without disturbing the meaning of text, sometimes it is possible to remove words, phrases, clauses and complete sentences. Summarizing the text is nothing but generating brief summaries of text without any repetitions. The limitation of statistical analysis of syntactic relations is the use of formal ontology and the deficiency of a standardized way of knowledge representation [1] [2] [3] [4]. Formal ontology does not support cognitive activity properties and also meaningful or sensible summarization of concepts. Knowledge representation in a traditional way is complex. And text summarization process may need different perspectives analysis of the input. The uniform representation enables grouping of knowledge from various domains into a single representation by using structural coordination. This is beneficial for the summary efficiency.

The text summarization problem can be modeled as Finite State Machine (FSM) [5]. It is a process which makes a single transition by joining state-transitions, induced by consecutive sentences.

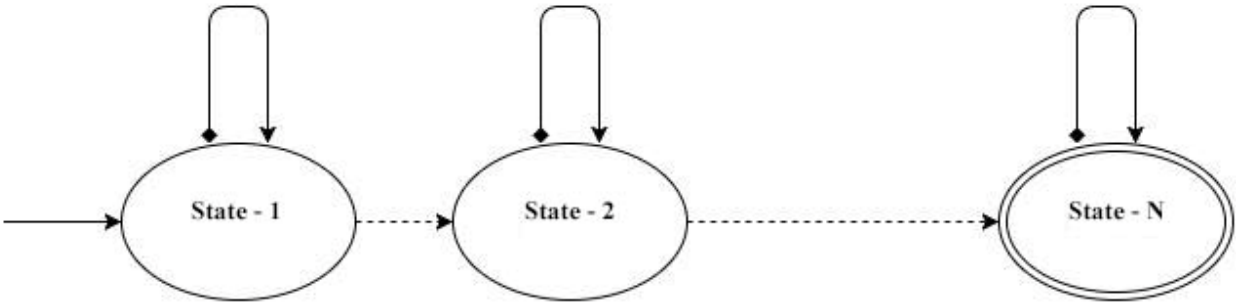

Figure 1: Text Summarization as a Finite State Machine

Two approaches are available to get the summary of the text: extraction and abstraction [6]. Extraction method concatenates extracts picked up from the corpus to form summary. Abstraction method

978-1-5386-2361-9/17/$31.00 ©2017 IEEE

generates new sentences from the information obtained from the corpus to form the summary. Extraction method is inappropriate for the multi-document summarization. Reason is the possibility of producing summaries which are biased with respect to some sources [7]. On the other hand, a little effort has been done in identifying the factors which affects the performance of each of the approaches in summarizing the evaluative documents. Evaluative documents are the ones which contains opinions and preferences, e.g. blogs or customer reviews.

II. LITERATURE SURVEY

The work which has been done on summarizing the single document is mainly on technical documents. The most cited technical paper on summarization is Luhn, 1958 [8]. This paper describes the research work done in 1950s at IBM. In 1958, Baxendale claimed that the sentence position provides an insight into its relative importance in the document [9]. In 1969, Edmundson illustrated a structure that constructs document extracts [10]. In 1995, Kupiec et al. described a method which was derived from Edmundson , which was capable to be trained from data [11]. Each sentence is worthy or not will be categorized by the classification function, by applying Naive-*Bayes classifier*.

….(1)

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

$$P(c|x) = P(x_1|c) \times P(x_2|c) \times \ldots \times P(x_n|c) \times P(c)$$

where,

P(c|x): posterior probability
P(x|c): likelihood
P(c): class prior probability
P(x): predictor prior probaility.

The features were agreeable to (Edmundson, 1969). But two more additional features were included, *sentence length* and the *presence of uppercase words* [12]. They illustrate a system called DimSim, which uses the features term frequency *(tf)* and inverse document frequency *(idf)*. This system derives the *signature words* [13]. In 1999, Lin came out of the hypothesis that features are not depending on each other. As an alternate to Naïve-Bayes classifier he used decision trees to model sentence extraction problem. He studied bunch of features and examined the effect of them on the sentence extraction. The data used or this work is collection of texts which are publicly available and are categorized into range of topics. It was given by the evaluations called TIPSTER-SUMMAC [14, 15], and targeted towards information retrieval systems.

The approaches which were used previously were non-sequential and feature-based. Contrast to that Conroy and O'leary (2001) used Hidden Markov Model to model the sentence extraction problem from document, which is as shown in figure 2 [16]. The basic reason behind using sequential model is to account for dependencies (local) among sentences.

This model uses only three features:

- In document the sentence position.
- Terms count that the sentence contains.
- Given the document terms, the likeliness of sentence terms.

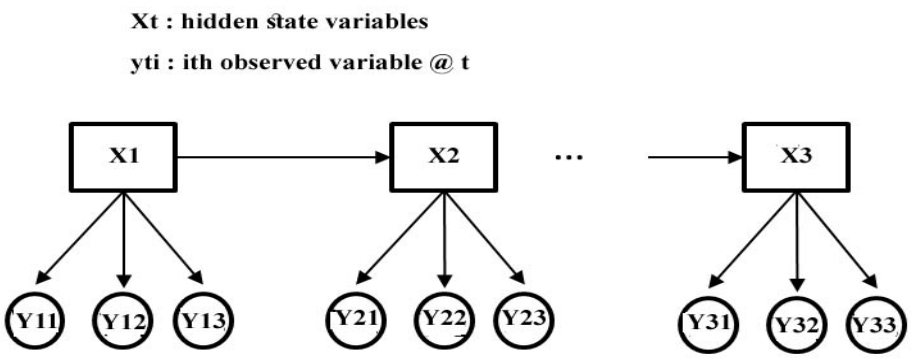

Figure 2: Hidden Markov Chain is modeled for Text Summarization

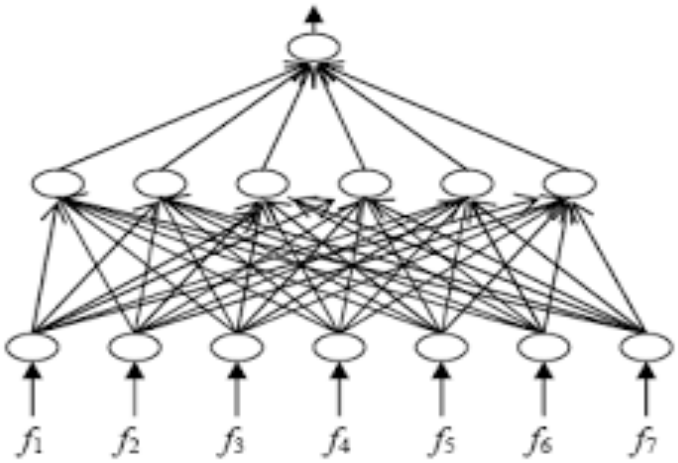

Figure 3: Feed Forward Neural Net with two hidden layers

In 2007 Svore et al. come up with a method based *neural nets* (figure 3). To handle the extractive summarization problem he used third part data sets. He trained the model using labels and features for each sentence of a document, so that it results in appropriate ranking of sentences in test document [17]. A pair-based neural network algorithm is designed to rank set of inputs which uses gradient descent technique for training, called RankNet (Burges et al., 2005). To find similarity score between human written emphasize and a sentence in the document they used ROUGE-1 [18] for the training data set.

III. IMPLEMENTATION

By mining key text fragments from text, extractive summaries are generated [19]. It uses statistical analysis of individual or mixed surface level features to find the location from where the sentences are to be extracted.

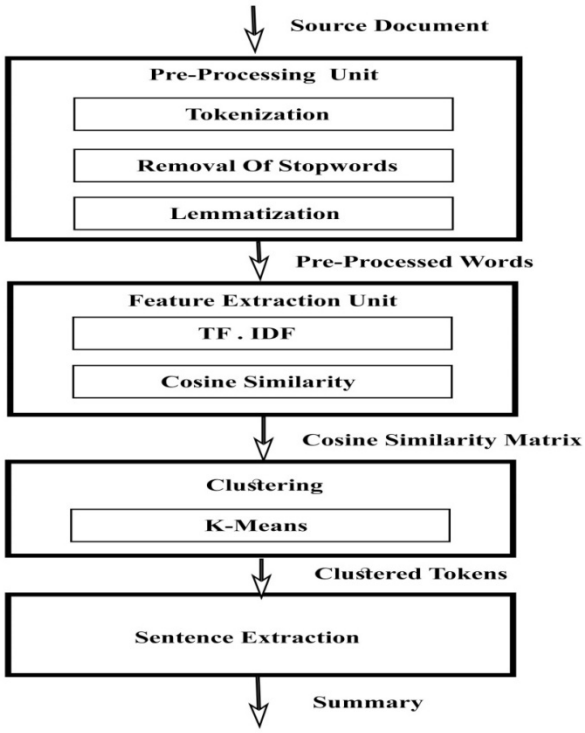

Figure 4: Implementation Flowchart

Extractive process [19] can be made into the following steps as shown in figure 4:

A. Preprocessing

Preprocessing can be viewed as original text structured representation. It basically includes:

1) Identification of sentence boundary. In English, each sentence ends with full stop (dot). This is used to identify the sentence.
2) Elimination of stop-words. Words which do not give any relevant information or words with no semantic meaning are eliminated.
3) Morphological analysis – stem or radix of a word gives its semantics. The purpose of morphological analysis is to get the stem or radix of words.

The features which have influence on the sentence relevance will be decided. To these features eights will be assigned using weight leaning technique. By applying feature-weight equation the final scores of each of the sentences will be determined. For final summary the highest ranked sentence will be selected.

B. Tokenization

Lexical analysis transforms the source document into a set of tokens. Each sentence will be divided into number of tokens. Non ASCII characters are removed since they do not carry implicit meaning. The final list of tokens is passed on to the next phase for further processing.

C. Removal of Stop words

These are the words which are strained from natural language sentences. Any group of words can be considered as stop words. Any word which does not contribute to the importance of a sentence can be removed and all those words are termed as stop-words. For the purpose of this project, stop words are listed from NLTK [20] to remove them from the document. Every word is compared to the group of stop words, which if matched is not considered as a new word in the vocabulary. So the vectors do not include stop words as a variable.

D. Term Frequency-Inverse Document Frequency (tf/idf)

Each element in the above sparse matrix must be assigned with a value. *tf/idf* rules are used [23] to find the weights of each word. *tf* is the count of appearance of the word in the whole document. *idf* is a measure of the importance of the word based on the rarity of its occurrence. After finding both values of a word in a sentence, its value will be computed as a product, as shown in equation 2.

E. Lemmatization

Grouping of various inflected forms of a word is called lemmatization. The purpose of this is to analyze them as a single entity[21]. In English, it is difficult to use a word in one single form. Often suitable equivalent forms of a word are used to grammatically avoid errors in a sentence. Lemmatization in the summarizer is employed as superior over the more primitive method - stemming [22].

F. Feature Extraction

Initially, the sentences are represented in the vectors of *n* dimension, where *n* is the count of words in the vocabulary of the document without the stop words. Each word is associated with a coefficient which represents the each word weight in the document. The collective weight of a word in a sentence defines the weight of the sentence. Sentence can be represented as a vector of n-dimension. For that a matrix of size *m* x *n* as shown in figure 5, is used where *m* is the count of sentences in the document and *n* represents the number of words in the vocabulary of the document without the stop words.

$$M = \begin{Bmatrix} a_{11} & a_{12} & \ldots\ldots & a_{1n} \\ a_{21} & a_{22} & \ldots\ldots & a_{2n} \\ & & . & \\ & & . & \\ a_{m1} & a_{m2} & \ldots\ldots & a_{mn} \end{Bmatrix}$$

Figure 5: Matrix representation of a Sentence

$$w_{i,j} = tf_{i,j} \times \log \frac{N}{df_i} \quad \ldots 2$$

Where,

tf_w: count of occurrences of i in j

df_i : count of documents containing i

N : total number of documents

Weights will be assigned to every word in a sentence. After weight assignment, we have a n-dimension vector for every sentence. The coefficients of words

are *tf/idf* values for words present in sentence and 0 otherwise.

G. Cosine Similarity

After calculating the *tf/idf* values of the document, the similarity of sentences is to be identified with respect to every other sentence. To do this, we use the trigonometric function cosine [24]. Every pair of vector in the *n*-dimensional space is related by angle θ (theta). Greater the angle lesser is the similarity of the sentence with each other. Cosine by definition varies from 1 to -1 as the angle varies from 0 to 180. Another easier way to find the cosine value of angles between the vectors is to find the dot product of the vectors and divide it by the product of their magnitudes.

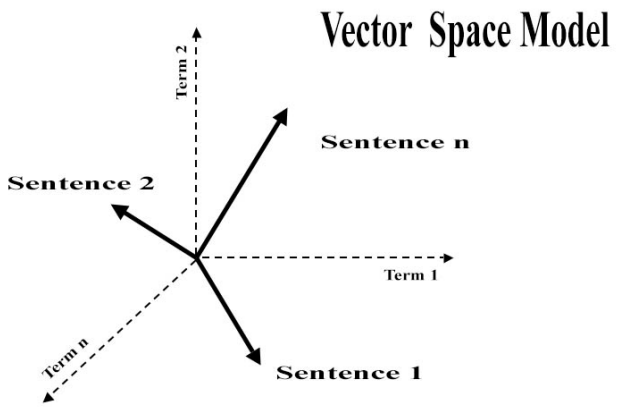

Figure 6: Cosine Similarity Measure

The cosine similarity of all the sentences is computed with respect to each other as shown in figure 6. The similarity of a vector with itself is made 0 instead of 1, since if the values are 1 then in the further steps the clustering goes wrong.

H. Clustering

Once the cosine similarity between vectors is identified, group similar sentences so that sentences are picked from each group. To do this, one of the machine learning methods of clustering called k-means [25, 26, 27] is used.

In data mining the popular cluster analysis is K-means clustering. It is vector quantization method. K-means clustering partitions n observations into k number of clusters. Based on the nearest mean value each observation fit in to the cluster. *K-means* is a unsupervised machine learning algorithm that takes number of clusters that has to be formed as input. Depending on the percentage of the summary required, the clusters numbers is calculated. In the presented work, the percentage is set to 30%.

I. Sentence Extraction and Summary Generation

In this phase, select a subset of sentences from each cluster. Since each cluster represents a single idea, one sentence from each would be enough to convey most of the message in that cluster. If the number of clusters is less than the number of ideas in the document, a single cluster may represent more than one idea. If a single cluster has a large percentage of sentences, the system picks more than one sentence from a single cluster. The sentence to be chosen from each cluster can either be random or can be ranked depending on parameters such as *tf/idf* values, number of words and so on. In the current implementation, one sentence from each cluster is chosen and the amalgamation of all chosen sentences forms the summary.

J. Evaluations

Evaluation of the resulted summary of the automatic text summarization systems is not an easy job. Text summarization is one of the hard and complex processes in natural language processing techniques, and the evaluation of such a summary produced is really hard. Usually human written summaries are more reliable and are always most concise. The goal of a summarizer must be to output a summary which is very similar to a human written summary. Considering the challenges of evaluating a summary, humans are needed for evaluating a summary.

The process of making a summary beneficial is an elusive property. It requires at least two properties of the summary which are to be measured to evaluate the summaries generated by the summarization systems:

1) Compression Ratio: compared to the original document how much shorter the summary

$$C\,R = \frac{length\ of\ Summary}{length\ of\ Full\ Text}$$

2) Retention Ratio: the amount of retained information

$$R\,R = \frac{information\ in\ Summary}{information\ in\ Full\ Text}$$

Retention Ratio is also called as Omission Ratio[28]. Evaluation of summarization systems must address both of these properties. Automatic text summarization systems are classified as intrinsic and extrinsic [29].

1) Intrinsic Evaluation

It is done by comparing the summary with a standard summary generated by humans. It is basically concetrated on the information of summaries and coherence of summaries.

2) Summary Information

There are two ways to measure how much information is preserved in the generated summary from the source document:

- Compare the condensed summary with the text.
- Compare the condensed summary with a reference summary.

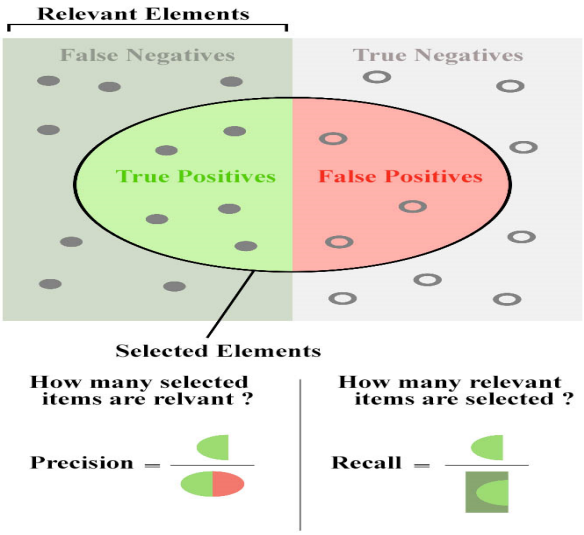

Figure 7: Precision Vs Recall

K. Summary Coherence

The generated summaries using extraction-based methods suffer from out of context parts, resulting in coherence problem. One method for summary coherence measure is, let summary sentences be ranked or graded for coherence, and then compare those grades with the scores for reference summaries.

L. Precision and Recall

These measures are standard for Information Retrieval system as shown in figure 7 and are often combined in F1-score [30]. They are not proficient enough to distinguish among many possible summaries.Different summaries with different contents may have similar F1-Scores.

M. Extrinsic Evaluation

It measures the effectiveness and acceptability of the summaries which are generated. Evaluation is in the form of relevance assessment, relevance feedback of query expansion in a search engine, question answering system, or reading comprehension. To what level it is promising to follow the summaries is also possible to measure. Other tasks which are measurable are, gathering the information from a huge document, the amount of time and effort it takes to post-edit the summary generated by the machine for some precise purpose, or the impact of summarization systems.

IV. RESULTS AND ANALYSIS

For the comprehensive evaluation, the proposed method is compared with brute force method. Results are computed on the basis of precision, recall and F1-score using an automatic evaluation toolkit ROUGE. The CNN dataset corpus is used for evaluation of the proposed architecture Vs brute force architecture.

Precision

Results on Precision are presented in Table and Figure 8. It is noticed that the brute force method performs poorly on precision in comparison with the proposed method. This can be attributed that the brute force method retrieves majority of sentences in the parent document. This poorly archives the intended goal of content reduction in the parent document. The proposed method performs only fairly on precision. The process of sentence selection from each cluster requires to be improvised.

TABLE I : PRECISION SCORE

Document	Precision	
	Proposed	***Brute-Force***
Article-1	0.2	0.04
Article-2	0.21	0.06
Article-3	0.44	0.10
Article-4	0.34	0.03
Article-5	0.17	0.013
Article-6	0.49	0.17
Article-7	0.33	0.03
Article-8	0.25	0.02
Article-9	0.12	0.01
Article-10	0.29	0.15

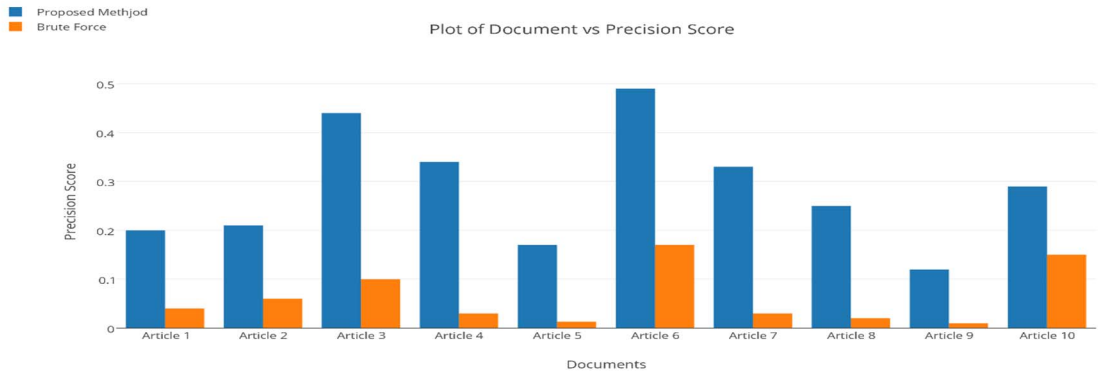

Figure 8: Precision Vs Documents

Recall

Results on Recall are presented in table 2 and figure 9. It is observed that the brute force method performs comparatively well and is in par with the proposed method. This can again be attributed to the fact that the brute force method retrieves majority of sentences in the parent document. The proposed method manages to recall only around 50% of required sentences. Hence, it can be inferred that superior clustering algorithms need to be explored.

TABLE II. RECALL SCORE

Document	**Recall**	
	Proposed	**Brute-Force**
Article-1	0.30	0.1
Article-2	0.27	0.2
Article-3	0.41	0.39
Article-4	0.31	0.31
Article-5	0.17	0.21
Article-6	0.3	0.19
Article-7	0.45	0.4
Article-8	0.42	0.21
Article-9	0.12	0.16
Article-10	0.29	0.12

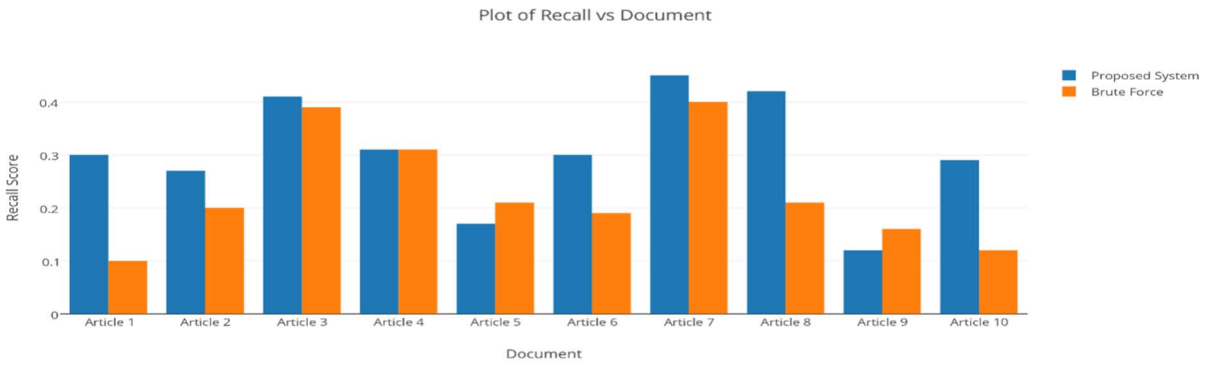

Figure 9: Recall Vs Documents

F1-score

Results on Fi-score are presented in table 3 and figure 10. The F1 score provides an insight into the overall performance of the summarizer. The brute force method performs poorly primarily due to its precision score. The proposed system performs better but still requires improvements as a product in real time.

TABLE III F1 SCORE

Document	F1 score	
	Proposed	Brute-Force
Article 1	0.12	0.0285
Article 2	0.1181	0.04615
Article 3	0.2122	0.07959
Article 4	0.16215	0.02735
Article 5	0.085	0.00355
Article 6	0.1860	0.08972
Article 7	0.19038	0.0279
Article 8	0.1567	0.01826
Article 9	0.06	0.00941
Article 10	0.145	0.06666

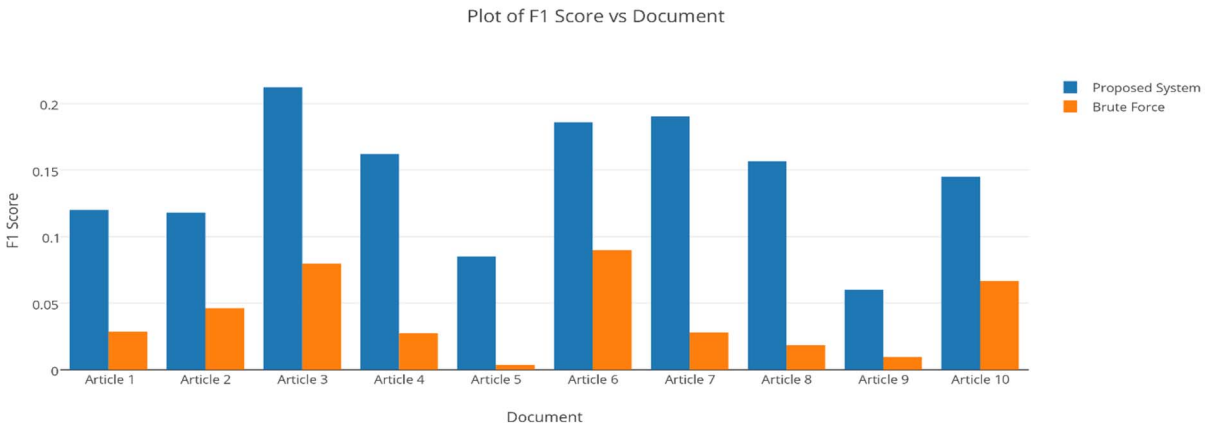

Figure 10: F1 Vs Documents

IV. CONCLUSIONS AND FUTURE WORK

The explosive increase in the size of the World Wide Web has made huge availability of text data. This work presents a method to achieve condensation of textual data, thus resolving the problem of redundancy and inaccuracy in a document. Proposed system performance is mediocre in comparison to commercial text summarizers. Through the course of evaluation, it is observed that k-means is a primitive clustering algorithm and advanced clustering algorithms are to be applied for superior results. Also, the process of sentence extraction from each

cluster must not be a random process but requires devising a specific algorithm.

REFERENCES

[1]. Jones, Karen Sparck, 'What might be in a summary?', Information retrieval 93, 1993, page: 9-26.

[2]. Endres-Niggemeyer, Brigitte, Summarizing Information: Including CD-ROM 'SimSum', Simulation of Summarizing, for Macintosh and Windows. Springer Science & Business Media, 2012.

[3]. Zhou, Liang, and Eduard H Hovy, 'On the Summarization of Dynamically Introduced Information: Online Discussions and Blogs', AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs. 2006.

[4]. Mani, Inderjeet, 'Automatic summarization', Vol. 3. John Benjamins Publishing, 2001.

[5]. Salton, Gerard, et al. 'Automatic analysis, theme generation, and summarization of machine-readable texts', Information retrieval and hypertext. Springer US, 1996. 51-73.

[6]. Hahn, Udo, Inderjeet Mani, 'The challenges of automatic summarization' , Computer 33.11, 2000, page: 29-36.

[7]. Barzilay, Regina, Michael Elhadad, 'Using lexical chains for text summarization', Advances in automatic text summarization, 1999, page: 111-121.

[8]. Luhn, Hans Peter, 'The automatic creation of literature abstracts', IBM Journal of research and development 2.2, 1958, page: 159-165.

[9]. Lin, Chin-Yew, Eduard Hovy, 'Identifying topics by position', Proceedings of the fifth conference on Applied natural language processing. Association for Computational Linguistics, 1997.

[10]. Edmundson, Harold P, 'New methods in automatic extracting', Journal of the ACM (JACM) 16.2, 1969, page: 264-285.

[11]. Kupiec, Paul H, 'Techniques for verifying the accuracy of risk measurement models', The Journal of Derivatives, 3.2, 1995.

[12]. Larsen, Bjornar, Chinatsu Aone, 'Fast and effective text mining using linear-time document clustering', Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 1999.

[13]. Miller, George A, 'WordNet: a lexical database for English', Communications of the ACM, 38.11, 1995, page: 39-41.

[14]. Hovy, Eduard, Chin-Yew Lin, 'Automated text summarization and the SUMMARIST system', Proceedings of a workshop on held at Baltimore, Maryland, Oct 13-15, 1998. Association for Computational Linguistics.

[15]. Mani, Inderjeet et al, 'The TIPSTER SUMMAC text summarization evaluation', Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 1999.

[16]. Conroy, John M, and Dianne P O'leary, 'Text summarization via hidden markov models', Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2001.

[17]. Svore, Krysta Marie, Lucy Vanderwende, Christopher JC Burges, 'Enhancing Single-Document Summarization by Combining RankNet and Third-Party Sources', EMNLP-CoNLL-2007.

[18]. Lin, Chin Yew, 'Rouge: A package for automatic evaluation of summaries, Text summarization branches out', Proceedings of the ACL-04 workshop. Vol. 8. 2004.

[19]. Gupta Vishal, Gurpreet Singh Lehal, 'A survey of text summarization extractive techniques', Journal of Emerging Technologies in Web Intelligence, 2.3, 2010, page: 258-268.

[20]. Bird, Steven, 'NLTK: the natural language toolkit', Proceedings of the COLING/ACL on Interactive presentation sessions. Association for Computational Linguistics, 2006.

[21]. Korenius, Tuomo et al, 'Stemming and lemmatization in the clustering of finnish text documents', Proceedings of the thirteenth ACM international conference on Information and knowledge management. ACM, 2004.

[22]. Jivani, Anjali Ganesh, 'A comparative study of stemming algorithms', International Journal of Comp. Tech. Appl, 2.6, 2011, page: 1930-1938.

[23]. Aizawa, Akiko, 'An information-theoretic perspective of tf–idf measures', Information Processing & Management, 39.1, 2003, page: 45-65.

[24]. Dehak, Najim et al, 'Cosine Similarity Scoring without Score Normalization Techniques', Odyssey, 2010.

[25]. Wagstaff, Kiri et al, 'Constrained k-means clustering with background knowledge', ICML, Vol. 1, 2001.

[26]. Hartigan, John A, Manchek A Wong, 'Algorithm AS 136: A k-means clustering algorithm', Journal of the Royal Statistical Society. Series C (Applied Statistics), 28.1, 1979, page: 100-108.

[27]. Kanungo, Tapas et al, 'An efficient k-means clustering algorithm: Analysis and implementation', Pattern Analysis and Machine Intelligence, IEEE Transactions on 24.7, 2002, page: 881-892.

[28]. Grishman, Ralph et al, 'Cross-lingual information extraction and automated text summarization', Multilingual Information Management: Current Levels and Future Abilities, 1999, page: 14.

[29]. Saracevic, Tefko, 'Evaluation of evaluation in information retrieval', Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1995.

[30]. Amati, Gianni, and Cornelis Joost Van Rijsbergen, 'Probabilistic models of information retrieval based on measuring the divergence from randomness', ACM Transactions on Information Systems (TOIS), 20.4 2002, page: 357-389.