International Journal of Database Theory and Application
Vol.10, No.8 (2017), pp.83-90
http://dx.doi.org/10.14257/ijdta.2017.10.8.08

Machine Learning Approach for Text Summarization

Amita Arora[1],Akanksha Diwedy[2],Manjeet Singh[3] and Naresh Chauhan[4]

[1]*Department of Computer Engineering, YMCA University of Science and Technology,Faridabad, India*
[2]*Student (BTech IT), YMCA University of Science and Technology,Faridabad, India*
[3]*Department of Computer Engineering, YMCA University of Science and Technology, Faridabad, India*
[4]*Department of Computer Engineering, YMCA University of Science and Technology, Faridabad, India*
amita.arora@gmail.com, akanksha.diwedy29@gmail.com
mstomer2000@yahoo.com,nareshchauhan19@gmail.com

Abstract

With the abundance of interminable text documents, providing summaries can help in retrieval of relevant information very quickly. The technique is to extract those sentences from the document that contain important information. This paper presents the results of our research on extractive summarization with a method based on Support Vector Machines (SVMs). The SVMs are trained using DUC-2002 dataset and the importance of sentences is judged on the basis of salient features. To evaluate the performance of our system, comparisons are conducted with two existing methods. ROUGE scores are used to compare the system generated summaries with the human generated summaries, and the experimental results show that our system's performance achieved high metrics.

Keywords*: Support Vector Machines, Machine Learning, Extractive summarization*

1. Introduction

Automatic text summarization is a process of making a consistent summarized document that keeps the most important points of original document. It is a method for data reduction which enables users to reduce the amount of text that must be read to gather the essential information. Summarization helps user to find meaningful and relevant information from large documents. It plays a significant role in information retrieval and information gathering.

Since the advent of text summarization, multiple techniques have been proposed for generating summaries in such a way that computer generated summary are similar to the human generated summary. Extractive summarization is one such approach that focuses on assigning scores [1] to sentences in the document based on certain predefined features. The extractive summary generated is a subset of the sentences from the original document. The features include use of proper nouns and word sense [2] ,linguistic and statistical features, such as position [Marcu, 1997] and syntactic features [Pollock and Zamora, 1975]. Each sentence feature has its own contribution for the relevant judgment of the sentence.

Multiple machine learning techniques have been employed for automatic text summarization, such as Bayesian classifiers [5], decision tree. However, most of these methods tend to overfit the training data when high dimension feature spaces are given. Support Vector Machines [5] are effective even with a high dimension feature space. In

ISSN: 2005-4270 IJDTA

this paper, we present the results of Single Document Summarization technique based on SVMs.

A. Support Vector Machines

Support Vector Machines performs classification tasks by constructing an optimal separation, referred to as hyperplanes in a multidimensional space that separates the training dataset items having different class labels as shown in Fig. 1.

For explanation purpose, we consider a linearly separable training data were each sample has feature x and label $y \in \{-1, 1\}$. -1 denotes negative class whereas 1 indicates positive class.

Therefore the training data is of the format $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$.

The maximum margin hyperplane,which resides equidistance from respective class support vectorsis expressed as

$$\vec{w}.\vec{x} + b = 0$$

Where $\vec{w}$ is the weight vector that is normal to the hyperplane and b is the scalar bias.

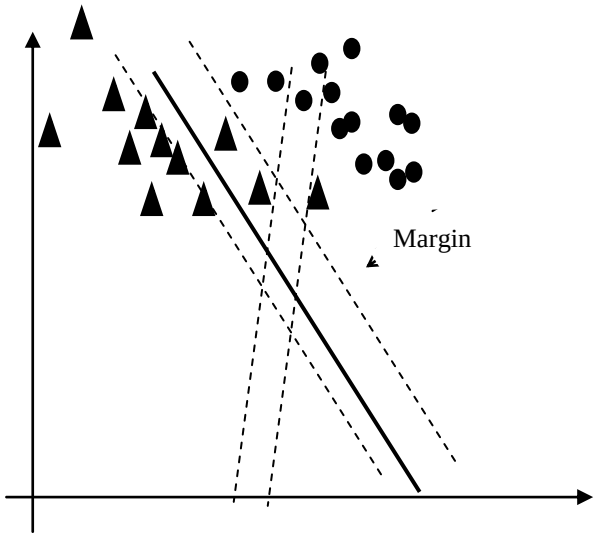

Figure 1. Optimal Decision Surface with Margin and Non-optimal Decision Surface

The objective is to maximize the distance between the parallel hyperplanes that separate the two classes of data having the maximum-margin hyperplane lying in between planes. Therefore, we have to minimize $\|w\|$:

$$min_{w,b} \frac{1}{2} ||w||^2$$

Subject to the equations :

$$\vec{w}.\vec{x} + b \geq 1$$
$$\vec{w}.\vec{x} + b \leq -1$$

In case of non-linear training data, slack parameters that take the misclassification rate into account are introduced:

$$min_{w,b} \frac{1}{2} ||w||^2 + C \sum_{i=1}^{n} \xi_i$$

Such that $y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 - \xi_i, \ i = 1, \ldots, n$
and $\xi_i \geq 0$

Here, C is a hyperparameter that controls the amount of training error allowed.

The above quadratic programming problem's solution provides the discriminant function as:

$$g(x) = \sum_{i=1}^{l} \lambda_i y_i x_i \ . x + b$$

Non-linear classifiers can be created by applying the kernel trick to maximum-margin hyperplane. The polynomial kernel is as follows,

$$k(\vec{x}_i, y) = (\vec{x}_i y + 1)^2$$

The decision function can be written in form:

$$f(x) = \sum_{i=1}^{l} \lambda_i . y_i . K(\vec{x}_i, x) + b$$

In this paper, we have used a polynomial kernel with the given parameters:

Degree = 2
C = 0.0001

The paper is organized as follows. Section 2 reviews the other works related to automatic summarization. Section 3 introduces our model based on Support Vector Machines. Section 4 presents the experimental setting and evaluates the results of the proposed model. Finally, Section 5 concludes the paper.

2. Related Work

Automatic text summarization has been studied since 1950s[1]. This approach was implemented on technical papers and magazine articles which suggested using frequency to determine words that are descriptive of the topic of the document. Conventionally, the sentence features were studied individually. [9]used position and length as salient surface features. It was based on the observation that sentences located at the beginning of the document were most likely composed of important information. [19] used lexical indicators to determine the relevant information from documents. Whereas, [3] used features like uppercase words, length, position of words by using naïve-bayes classifier.[20] used cohesion chains to determine the sequence of associated words. Edmundson's [17] work led to machine learning approaches in summarization. He used a linear combination of features to weight sentences. Thomas et al. [18] designed a system for automatic keyword extraction for text summarization using hidden Markov model. Lin and Hovy[21] used decision trees and rich features where the text is portrayed in a predictable discourse structure. The approaches used in [12] and [14] made use of SVM model for extracting summary with different set of feature vectors in each approach.

The authors in this paper have also used support Vector Machines with a certain set of features which shows good performance when experimented with DUC 2002 dataset.

3. Proposed Approach

Figure 2 shows the proposed approach for generating extractive summary from plain text document using SVM classifier.

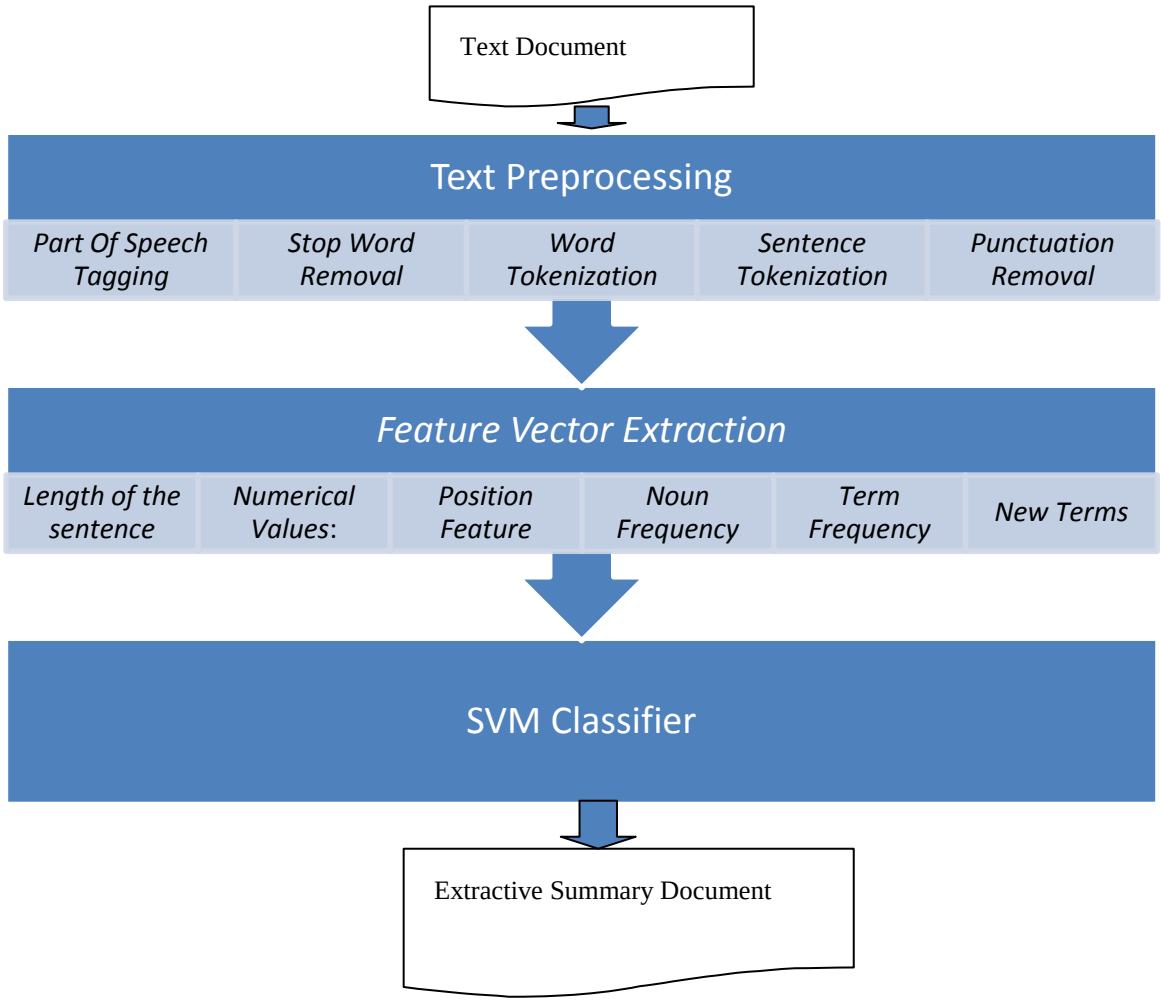

Figure 2. Proposed Approach for Extractive Summarization

The text for which summary is to be generated is preprocessed so that we can extract the feature vectors from that text. For this following steps are followed:

A. Pre Processing

1. *Sentence Tokenization*: We have divided the entire document into sentences. Each sentence is then processed individually.
2. *Word Tokenization*: Every sentence is then tokenized to generate tokens that are used to determine words and phrases.
3. *Stop Word Removal*: The stop words are removed from the list of tokens. The stop words are are taken from the NLTK corpus.
4. *Punctuation Removal*: All the punctuations are removed along with the stop words so that they are not included in the term frequency count.
5. *Part Of Speech Tagging* :Words of the document are tagged as nouns, adjectives, verbs. Tagging is done using NLTK pos-tagger.

B. Feature Vector Extraction

When the preprocessing of the document is completed, the feature vector of each sentence of the document is calculated. The feature vectors are combined to form the feature matrix of the entire document. The following features are taken into account:

1. *Position Feature*[7]: This feature is used to judge the importance of the sentence on the basis of its position in the document. For first or last sentence of document.

$$SenPos = 1$$

For the remaining sentences,

$$SenPos = \cos\{(sentencePosition - minTh) * \left(\frac{1}{maxTh} - minTh\right)\}$$

Where,

$$minTh = \frac{20}{100} * N * N$$

$$maxTh = \frac{40}{100} * N * N$$

N is total number of sentences in document.

2. *Numerical Values*: This feature consider the sentences that contain numerical values or figures.

$$Num\_val = \frac{Number\ of\ numerical\ terms}{length\ of\ the\ sentence}$$

3. *Term Frequency*: This feature helps to determine the importance of the sentence on the basis of the amount of terms that are frequently occurring in the sentence. This features captures the sentences that most likely contain relevant information.

$$Freq_{term} = \frac{Frequency\ of\ the\ term\ in\ the\ document}{Highest\ term\ frequency}$$

$$Sen\_Freq = \frac{\sum_{i=1}^{length} Freq_{term}}{length\ of\ the\ sentence}$$

4. *Noun Frequency*: Proper Nouns are calculated on basis of part-of speech tagging of each sentences.

$$Noun_{term} = \frac{Numbr\ of\ terms\ tagged\ as\ nouns}{length\ of\ the\ sentence}$$

5. *Lengthof the sentence*: Sentence length can help to determine the amount of content in the sentence.

$$Sen_{len} = \frac{Number\ of\ terms\ in\ the\ sentence}{length\ of\ longest\ sentence}$$

6. *New Terms*: This feature is used to calculate the number of unique terms in a sentences. It gives weightage to those sentences that contain new information.

$$New_{term} = \frac{Number\ of\ unique\ term\ in\ that\ sentence}{length\ of\ the\ sentence}$$

4. Experiment and Evaluation

The SVM classifier is trained using DUC-2002 dataset because it provides summaries along with the documents for comparison. For training, Binary classification problem is considered where every sentence is classified as important or not important. The class is judged on the basis of presence or absence of the sentence in the document's summary. The training dataset consists of 500 negative and 500 positive samples.

After the feature matrix of the document is calculated, it is given as an input to the trained classifier. SVM classifies the sentences as positive and negative. Only positively classified sentences are taken, and they are ranked according to the their distance from the maximum margin hyperplane. The top N sentences are chosen as the extractive summary for the document. N is varied according to length of summary that is to be generated. Randomly chosen documents were chosen by us for evaluation.

To evaluate the efficiency achieved by applying the proposed model, it is compared with the *Base* and *TextRank* summaries. We are using ROUGE evaluation package (Recall Oriented Understudy for Gisting Evaluation) for the evaluation of summaries. Recall

based score are used to compare system generated summary with one or more human generated summaries. Unigram matching is found to be the best indicator for evaluation. We are using ROUGE-1 scores which are computed as division of count of unigrams in relative that appear in system and count of unigrams in reference summary.Table 1 represents the recall, precision and F-score of the documents calculated by taking human generated summary as reference.

Table 1. Experimental Scores

Document	Recall	Precision	F-Score
1	0.63063	0.33333	0.43614
2	0.47253	0.43434	0.45263
3	0.48148	0.33121	0.39245
4	0.57843	0.472	0.51982
5	0.69524	0.39891	0.50694
6	0.56	0.31285	0.40143
7	0.56364	0.47692	0.51667
8	0.59804	0.31606	0.41356
9	0.50495	0.65385	0.56983
10	0.58879	0.33158	0.42424
11	0.66981	0.29461	0.40922
12	0.76636	0.36771	0.49697
13	0.54808	0.43846	0.48718
14	0.61538	0.27948	0.38438
15	0.65217	0.30303	0.41379
16	0.625	0.38462	0.47619
18	0.53398	0.3624	0.43307
19	0.45361	0.48889	0.47059

The results obtained are compared with Base Line summaries and TextRank summaries. The graph shown in Figure 3 indicates the average Recall scores for the summaries of the three models. High average recall metrics is observed for our proposed model.

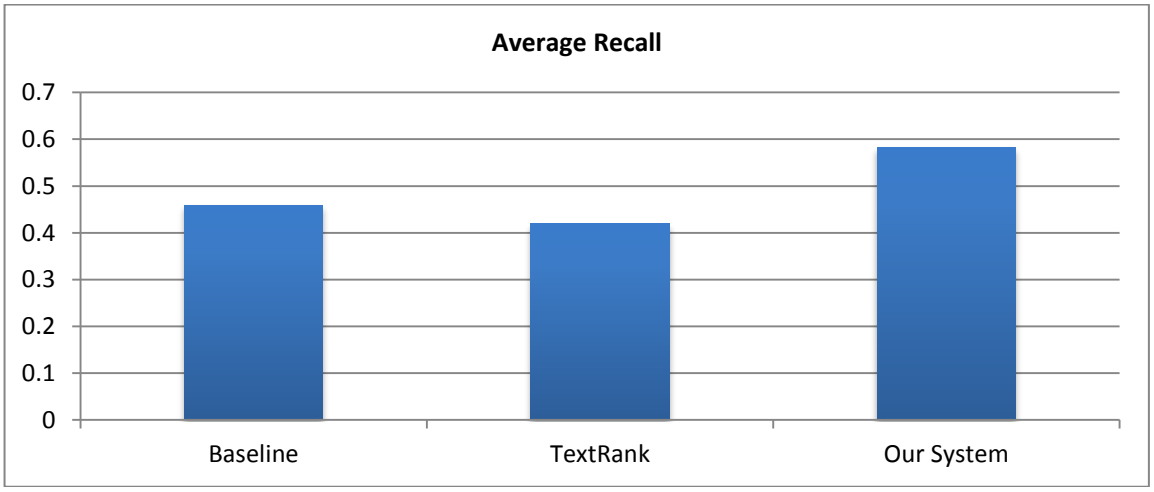

Figure 3. Result Evaluation of Our Approach

5. Conclusion and Future Work

Summarizing a text automatically with a good accuracy is tedium. The experimental results showed that our proposed system based on Support Vector Machines achieved good performance with high metrics values. To further improve the accuracy, experiments will be conducted with additional combination of features and incorporation of ontologies for better text summarization. In future we plan to experiment our approach for multi-document summarization.

References

[1] H. P. Luhn, "The Automatic Creation of Literature Abstracts", Presented at IRE National Convention, **(1958)**, pp. 159-165.
[2] Moschitti and R. Basili, "Complex linguistic features for text classification: A comprehensive study", In ECIR, **(2004)**; Sunderland, UK.
[3] J. Kupiec, J. Pedersen and F. Chen, "A Trainable Document Summarizer", Proc. of the 18th ACM-SIGIR, **(1995)**.
[4] S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach (2nd ed.)", **(2003)**.
[5] V. Vapnik, "The Nature of Statistical Learning Theory, **(1995)**; New York.
[6] S. P. Singh, A. Kumar, A. Mangal and S. Singhal, "Bilingual Automatic Text Summarization Using Unsupervised Deep Learning", **(2016)**.
[7] T. Joachims, "Transductive inference for text classification using support vertor machines", Proc. 16th International Conf. on Machine Learning, **(1999)**; Morgan Kaufmann, San Francisco, CA.
[8] C. Aone, M. Okurowski and J. Gorlinsky, "Trainable Scalable Summarization Using RobustNLPandMachineLearning", Proc. of the 17th COLING and 36th ACL, **(1998)**.
[9] D. Radev, "Text Summarization Tutorial", Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR), **(2000)**.
[10] C.Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries" Proceedings of Workshop on Text Summarization of ACL, **(2004)**; Spain.
[11] R. B. Yates and B. R. Neto, "Modern Information Retrieval", (1st ed.)Addison Wesley, **(1999)**.
[12] T. Hirao, H. Isozaki, E. Maeda and Y. Matsumoto, "Extracting Important Sentences with Support Vector Machines", COLING: The 19th International Conference on Computational Linguistic, **(2002)**.
[13] T. Hirao, Y. Sasaki, H. Isozaki and E. Maeda, "NTT's Text Summarization system for DUC", Proceedings of the Workshop on Automatic Summarization, **(2002)**; Philadelphia, PA.
[14] V. D. Thanh, V. T. Hung, H. K. Hung and T. Q. Huy, "Text classification based on SVM and text summarization", Proceeding of the ACIS, **(2014)**.
[15] J. J. Pollock and A. Zamora, "Automatic Abstracting Research at Chemical Abstracts Service", Journal of Chemical Information and Computer Sciences, vol. 15, no. 4, **(1975)**, pp. 226-232.
[16] J. Larocca, A. Neto, A. Freitas, A. Celso and A. Kaestner,"Automatic Text Summarization using a Machine Learning Approach".
[17] H. P. Edmundson, "New methods in automatic extracting", Journal of the ACM, vol. 16, no. 2, **(1969)**, pp. 264–285.
[18] J. R. Thomas, S. K. Bharti and K. S. Babu, "Automatic keyword extraction for text summarization in e-newspapers", Proceedings of the International Conference on Informatics and Analytics, ACM, **(2016)**.
[19] G. J. Rath, A. Resnick and T. R. Savage, "Comparisons of four types of lexical indicators of content", Journal of the American Society for Information Science and Technology, vol. 12, no. 2, **(1961)**, pp. 126-130.
[20] J. Morris and G. Hirst, "Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text", Journal of Computational Linguistics, vol. 17, no. 1, **(1991)**, pp. 21-48.
[21] C.-Y. Lin and E. Hovy, "Identifying topics by position", Proceedings of the Fifth conference on Applied natural language processing, **(1997)**; San Francisco, CA, USA.

Authors

Amita Arora is presently working as an Assistant Professor at Department of Computer Engineering in YMCA University of Science and Technology, Faridabad. She has ten years of experience in teaching. She has supervised six M, Tech Thesis. Her current research interests are Semantic Web, Information Retrieval, Natural language Processing. She has been teaching subjects like Analysis and Design of Algorithm, Compiler Design, Computer Graphics. She has published seven articles in International/National Journals and Conferences.

AkankshaDiwedy is presently a final year undergraduate student. She is pursuing BTech in Information Technology at YMCA University of Science and Technology, Faridabad. Her current research interests include Machine Learning, Natural Language Processing and Information Retrieval. She has published one article in National Conference.

Manjeet Singh is presently working as Professor at Department of Information Technology and Computer Application, YMCA University of Science and Technology, Faridabad. He has fifteen years of research and teaching experience and has supervised successfully 04 Ph.D and around 20 M.Tech theses. His current research interest includes Natural Language Processing, Semantic Web, Information Retrieval, Computer Networks, Ad-Hoc Networks. He has been teaching subjects like Artificial Intelligence, Soft Computing, Computer Networks, Compiler Design, Discrete Structures etc. He has published 47 article in International/National Journals and Conferences.

NareshChauhan is working as Professor at Department of Computer Engineering,YMCA University of Science and Technology, Faridabad. He has 24 years of experience in the field of embedded software development and software testing and teaching various subjects in the area of Software Engineering, Software Testing, Software Project Management, Operating Systems, Embedded Systems, Real Time Systems, OOPS. He has published 37 research papers in various national and international journals, and 52 research papers in various national and international conferences. He has guided successfully 3 PhD scholars and currently he is guiding 5 Ph.D. Scholars on the topics of Software Testing, Agile Software Development, Internet& Web Technology.

