IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 1, January 2018

Machine Learning Approach for Automatic Text Summarization Using Neural Networks

Meetkumar Patel[1], Adwaita Chokshi[1], Satyadev Vyas[2], Khushbu Maurya[3]

Student, Department of Computer Engineering, Gujarat Technological University, Ahmedabad, India[1]
Head of the Department, Department of Computer Engineering, Gujarat Technological University, Ahmedabad, India[2]
Professor, Department of Computer Engineering, Gujarat Technological University, Ahmedabad, India[3]

Abstract: Machine learning and deep learning, as we know, have started ruling over almost every field in the computing industry and so, has revolutionized the process of text summarization too. Automatic text summarization is an advancing realm of the natural language processing research in which concise textual summaries are generated from lengthy input documents. Extensive research has been carried out on how automatic summarization can be prosecuted through various extractive and abstractive techniques. In this paper, we address all the approaches to text summarization and present the modus operandi of an Architecture called Encoder-Decoder, under the machine learning approach. Moreover, we propose several novel implementation models for this architecture, in Keras and TensorFlow that consists of various machine learning and deep learning neural network libraries.

Keywords: Machine Learning, Text Summarization, Neural Networks, Deep Learning, Keras, TensorFlow.

I. INTRODUCTION

Text mining, to begin with, deploys some of the techniques of Natural Language Processing (NLP) such as Parts-Of-Speech (POS) tagging, N-grams, parsing, tokenization, etcetera to carry out text analysis. It includes tasks like automatic keyword extraction and text summarization. Particularizing furthermore, text summarization is the process of shortening a text document with a computer program, in order to generate a summary with the major points extracted out of the original document. The main idea behind text summarization, is to elicit large quantities of text to derive high-quality abridged information, which preserves its primary meaning withal. Various variables that are vital for the recapitulation of an intelligible summary generally include length, syntax and the style of writing. Text summarization methods can be roughly divided into two categories: extractive summarization and abstractive summarization [1].Extractive summarization mines key sentences or phrases from the source documents and clutch them to produce a summary without changing the source text. Here, the sentences are generally in the same order as in the text of the source document. On the contrary, abstractive summarization entails understanding the source text by using semantic methods to examine and construe the text. This approach targets to generate a more generalized summary that depicts the information in a succinct way, and usually requires advanced language generation and compression techniques [2].

At present, there is tremendous amount of data and information available on every platform; but a human hardly gets any time, even to sift through a newspaper. Going through heaps of documents and information can be challenging and time consuming at the same time. Without a proper summary or abstract, it can get really tedious, just to get the gist of what someone is talking about in a paper or a report. Hence, text summarization comes here to the save, as it minimizes the reading time by extracting salient details from a document and condensing it into a summary. In this way it becomes easy to quickly assess whether or not a document is worth reading and investing time into. This comes in very handy for analysts, business leaders, paralegals and academic researchers as they have to comb through numerous documents every day.

Amongst the methods briefed above, in this paper, we have discussed about the machine learning approach that uses artificial neural networks to generate summaries of arbitrary length articles. Specifically, the Encoder-Decoder recurrent neural network (RNN) architecture developed for machine translation has found out to generate promising results when applied to the problem of text summarization. The architecture consists of two neural networks working in parallel simultaneously - an encoder that takes the input sequence and produces a vector output and the decoder that takes the previous vector output as its input and generates the final output sequence. The paper covers a general overview of the approaches to text summarization initially; and then details on the Encoder-Decoder model of the machine learning approach along with its implementation in Keras using TensorFlow.

DOI 10.17148/IJARCCE.2018.7132

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 1, January 2018

II. RELATED WORK

Text summarization by the means of natural language processing, dates back to the late 1950s. The early text summarization techniques were based on statistical methods as published in [3], in 1958. These methods mainly involved selecting large blocks of text for generating relative and rational abstracts for the same. Furthermore, this work progressed to better and more persuasive results with the help of graph based ranking model for text processing as elaborated in [4] and with Maximal Marginal Relevance (MMR) criterion as detailed in [5]. Meanwhile, evaluation measures like Recall-Oriented Understudy for Gisting Evaluation (ROUGE) and Bilingual Evaluation Understudy (BLEU) were invented that determined how well an automatic summary covered the matter, present in an original text. Also, different datasets like the DUC series, Medline, TAC series etcetera were developed so that comparison and contrasting of various summarization methods would be possible.

Generally, extractive text summarization process was primarily used. But gradually, the practice of abstractive text summarization has brought a momentous change in this field. This might be because of the techniques that are being used under the categories of structure based methods and the semantic based methods that condenses a text more strongly than the extraction methods. It chiefly involves converting a text into pre-processed form before finally converting it to a summary, as explained in [6] for a particular Indian language. There are comparatively lesser works done in this process as it requires the usage of natural language generation technology, which is harder to develop.

The most thriving technique from all, has been the usage of deep neural networks that are tremendously powerful machine learning models and can achieve excellent performance in sequence learning, because of their parallel computation. The Recurrent Neural Network (RNN) [7][8] is a natural generalization of feedforward neural networks to sequences that computes a sequence of outputs on the base of a given set of inputs. This is rightly elucidated in [9] which also debunks how RNN proves to be inefficient eventually, because of the resulting long term dependencies which is why the concept of Long Short-Term Memory (LSTM), that learns problems with long range temporal dependencies has found out to be better. As far as our research work goes, we have used encoder-decoder architecture for text summarization using Keras and Tensorflow libraries for neural networks and DUC 2005, NewsIR '16 and NIPS conference articles for the dataset.

III. TEXT SUMMARIZATION APPROACHES

Based on the literature, text summarization approaches can be relegated into five types, namely statistical based, machine learning based, coherent based, graph based, algebraic based as shown in Figure

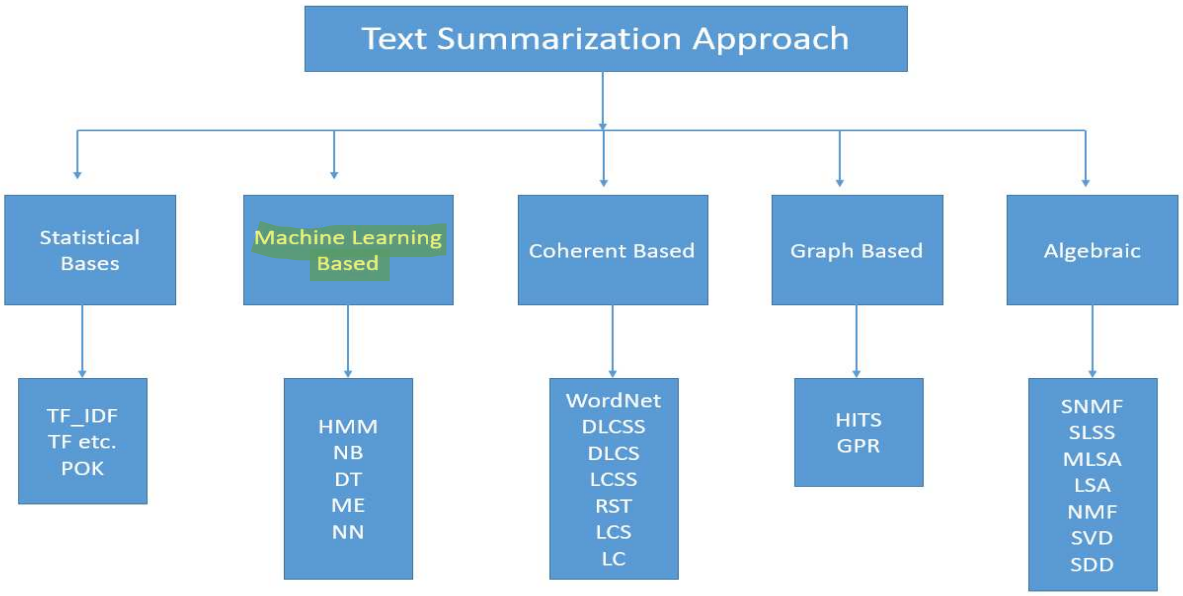

FIG. 1. TEXT SUMMARIZATION APPROACHES

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 1, January 2018

TABLE I
LIST OF ABBREVIATIONS USED IN CLASSIFICATION OF TEXT SUMMARIZATION APPROACHES

Abbreviations	Meaning
TF	Term Frequency
IF-IDF	Term Frequency-inverse document frequency
POK	Position of a Keyword
HMM	Hidden Markov Model
DT	Decision Trees
ME	Maximum Entropy
NN	Neural Networks
NB	Naïve Bayes
DLCSS	Direct lexical chain span score
DLCS	Direct lexical chain score
LCSS	Lexical chain span score
LCS	Lexical chain score
RST	Rhetorical Structure Theory
LC	Lexical chain
GPR	Google's Pagerank
HITS	Hyperlinked Induced Topic Search
MLSA	Meta Latent Semantic Analysis
SNMF	Symmetric nonnegative matrix factorization
SLSS	Sentence level semantic analysis
LSA	Latent Semantic Analysis
NMF	Non-Negative Matrix factorization
SVD	Singular Value Decomposition
SDD	Semi-Discrete Decomposition

1. ***Statistical Based Approach***

This approach is very simple and often utilized for keyword extraction from the documents. There is no predefined dataset required. In this approach we use word frequency, uppercase words, sentence length, keywords, position in complete text and phrase structure[10][11][12].

2. ***Machine learning Based Approach***

A trainable text summarizer can be obtained by the application of a trainable machine learning algorithm. Basically it is a feature dependent approach and we need annotated dataset to train the models [01]. There are various popular machine learning approaches namely, Decision trees [16][17], Nave Bayes [13][14][15], Hidden Markov Model[18][19][20], Maximum Entropy [21][22][23][24][25], Neural Networks [26][27][28], and Support Vector Machine [29][30][31] (SVM) etc.

3. ***Coherent Based Approach***

Basically, a coherent based approach deals with the cohesion relation between the words. There are various cohesion relations among elements in a text like reference, ellipsis, substitution, conjunction, and lexical cohesion [32]. We can see the classification of this approach in figure 1.

4. ***Graph Based Approach***

Graph based approach mainly introduces two popular approaches for text summarization namely, Google's PageRank (GPR)[33][35][36] and Hyperlinked Induced Topic Search (HITS)[33][34]. Google PageRank is used by Google Search to rank websites in their search engine results. GPR is a way of measuring the importance of webpages[02].

5. ***Algebraic Approach***

In algebraic approach we use algebraic theories like, matrix, Eigen vectors, transpose of matrix, etc. There are various algorithms used for text summarization using this approach like LSA- Latent Semantic Analysis[37][38][39], SNMF-Symmetric nonnegative matrix factorization[42], MLSA-Meta Latent Semantic Analysis[40][41], SLSS-Sentence level

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 1, January 2018

semantic analysis[42], NMF-Non Negative Matrix factorization[43], SVD-Singular Value Decomposition[44], SDD-Semi Discrete Decomposition[45], etcetera.

IV. METHODOLOGY

A. *Encoder-Decoder LSTM Architecture*

This architecture is a way of organizing recurrent neural networks for sequence prediction problems, which have multiple number of inputs, outputs, or even both. The architecture mainly involves two major components which are: an encoder and a decoder. Encoder, chiefly reads the entire input sequence in a go and encodes it into an internal representation, sometimes in a fixed-length vector, also called the context vector. The decoder reads the encoded input sequence resulted from the encoder and generates an output sequence. Both the sub models are trained in parallel and at the same time.

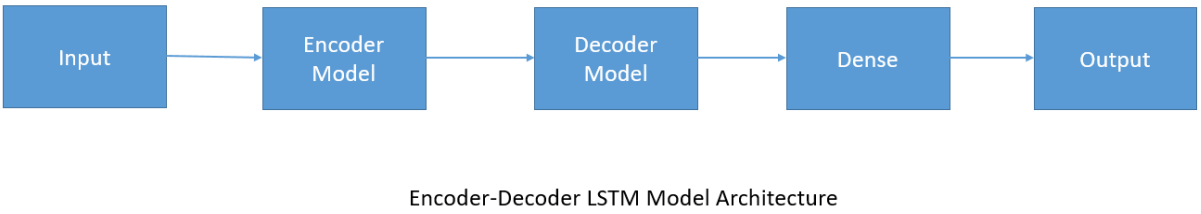

FIG. 2. ENCODER-DECODER LSTM MODEL ARCHITECTURE

Encoder-Decoder LSTM architecture was designed specifically for sequence to sequence problems. Mainly this model was developed for natural language processing problems where it demonstrated state-of-the-art performance, particularly in the area of text translation called statistical machine translation. State-of-the-art models look up embeddings of their inputs and pass them through a bidirectional recurrent neural network to produce the initial hidden state.

B. *Text Summarization Encoders*

Encoders, in the model, are like that brain of a human where major complexities dwell. This is because the encoders have the key function of capturing the tenor of the original document and generating an intermediate representation as an output that can be fed to the decoders for the final output. There are several types of encoders that can be used; albeit some of them are more common in usage such as bidirectional recurrent neural networks (Bi-RNNs). LSTMs in particular, are generally preferred while using Bi-RNNs for text summarization. In cases where RNNs are used in the encoder, word embedding is often used to offer a distributed representation of the words[52][49]50]51].

C. *Text Summarization Decoders*

The decoders are responsible for the generation of words in the output sequence, when two sources of information are available: a) a Context Vector, which is an encoded representation of the source document in a vector format, resulted as an output of the encoder and b) a Generated Sequence, which is an already abstracted word or sequence of words. Specifying for a simple Encoder-Decoder Architecture, the Context Vector may be a fix-length encoding structure or a more comprehensible form that might be filtered by the means of an attention mechanism. The distributed representation of each word generated via a word embedding is provided to the Generated Sequence. The process generally involves providing the model with a special start-of-sequence token, in order to generate the first word; and also an end-of-sequence token to stop the process [53][54].

V. DATASETS

We have used three different datasets for training our models:

A. DUC 2005

While many datasets are available that approximate the idea of summaries by using abstracts or headlines, this classic dataset from the Document Understanding Conference [11] contains multiple human generated labels per document that were explicitly constructed as summaries. The DUC dataset contains 500 documents of which we used 432 news articles with unique beginnings, with 4 to 8 model summaries each. While these are high-quality gold-standard summaries, the dataset is too small to use as an exclusive training set. Thus it is often used as a test set only; we used it both for training in our smaller models and for testing for all models.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 1, January 2018

B. NewsIR '16

This is a "Signal Media One-Million News Articles Dataset" [46] dataset that contains about a million news articles and their titles collected from a wide range of sources. We filtered this dataset for particular news articles category, ("media-type": "News") in order to collate it with the DUC dataset, leaving 572,154 article-title pairs.

C. NIPS articles, abstracts & titles

This dataset contains all published papers from the Neural Information Processing Systems (NIPS) conference. [47] We used the article's titles and abstracts; and extracted them to create two sub-datasets: nips-article-abstract and nips-abstract-headline. As cited by its author, it contains conference papers and articles, "ranging from the first 1987 conference to the current 2017 conference".

VI. IMPLEMENTATION MODELS

In this section we have illustrated, how the Encoder-Decoder architecture is implemented using various models for text summarization in the Keras deep learning library. Here we are going to elaborate on four models: General Model, One-Shot Model, Recursive Model A and Recursive Model B.

A. *General Model*

General model is a simple realization of the model. It involves an Encoder with an Embedding input followed by an LSTM hidden layer. That hidden layer produces a fixed length representation of the source documents. The decoder reads the representation along with the Embedding of the last generated word and uses these inputs to generate each word in the output. The following figure describes the architecture of the general text summarization process.

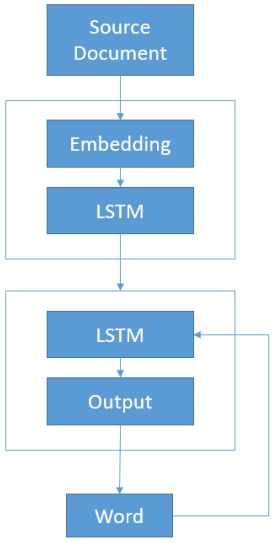

FIG. 3. GENERAL TEXT SUMMARIZATION MODEL IN KERAS

The general text summarization model has some drawbacks that can be explained as follows: Keras does not allow recursive loops where the output of the model is fed as an input to the general model automatically. Basically it means that the general text summarization model cannot be directly implemented using Keras, but can be implemented by using a more flexible platform like TensorFlow.

B. *One-Shot Model*

One-shot model as the name suggests, generates an entire output sequence in a single shot. In this model, the decoder solely uses the context vector to generate the output sequence. The figure below depicts the architecture of the One-shot text summarization model. One-shot model imposes a heavy burden on the decoder. Additionally, it becomes challenging for the decoder to generate a coherent output sequence, as it must choose the words with their order, without any sufficient context.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 1, January 2018

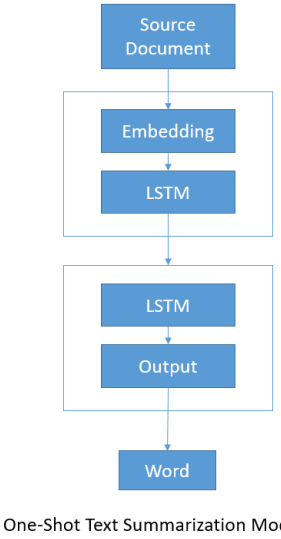

FIG. 4. ONE SHOT TEXT SUMMARIZATION MODEL

C. *Recursive Model A*

Recursive model A is a second alternative model of the general text summarization model. It generates a single word forecast and calls it recursively. In this model, the decoder uses the context vector and the distributed representation of all words generated so far, as input in order to generate the next word.

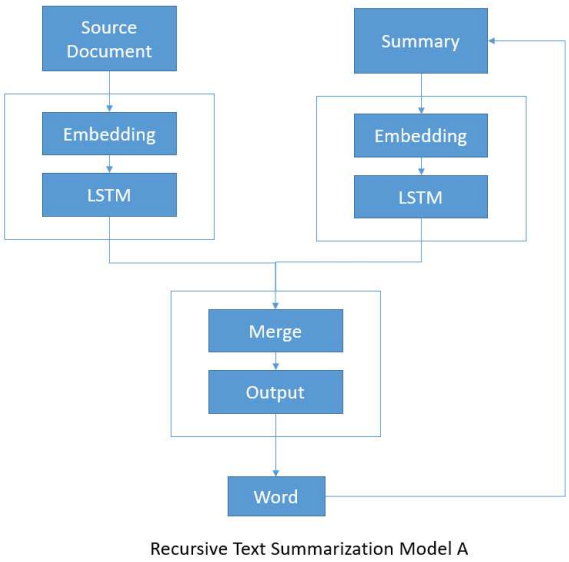

FIG. 5. RECURSIVE TEXT SUMMARIZATION MODEL A

In this model, the output summary is built by recursively calling the model and appending the outputs with the previously generated word repetitively. This model is better because the decoder is provided with an option to use the previously generated words and the source document as a context for generating the next word. This model burdens the merge operation and the decoder to infer where it is up to, in generating the output sequence.

D. *Recursive Model B*

In the Recursive Model B, the Encoder generates a context vector representation of the source document. This representation is further fed to the decoder at each phase of the generated output sequence. This activity helps the decoder in building up the same internal state as was used to generate the words in the output sequence [55].

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 1, January 2018

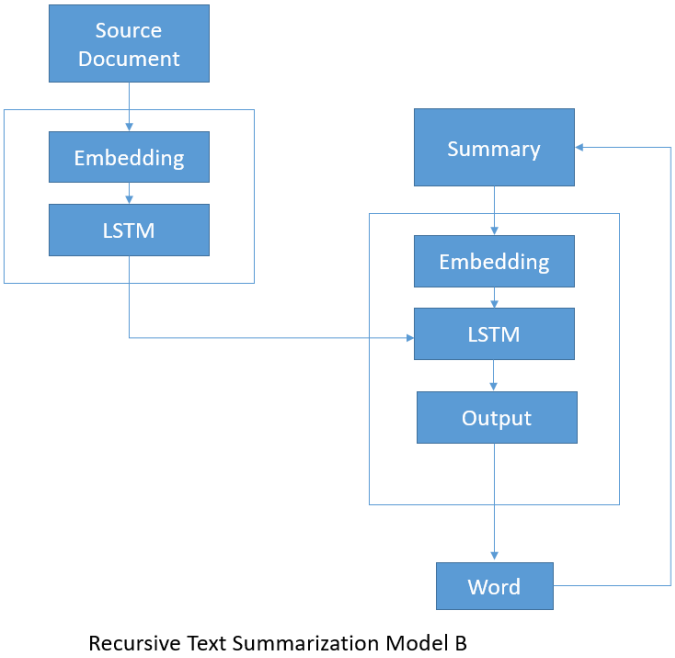

FIG. 6. RECURSIVE TEXT SUMMARIZATION MODEL B

As shown in the figure above, this process is then repeated by calling the model over and over again for each word in the output sequence until a maximum length or end of sequence token is generated.

VII. CONCLUSION AND FUTURE RESEARCH

In this research work, we have explored the machine learning approach to text summarization and have presented a range of novel models using neural networks. Because of its wide range of applications, extensive investigation has already been carried out on extractive text summarization techniques but because of its complexity, comparatively less research has been done on abstractive text summarization techniques. In our proposal, we have employed different models with variety of datasets, for trainable text summarizers using Keras and TensorFlow libraries and have analyzed all the results. In future we would like to continue our research by adding diverse sentence simplification techniques for aiming better results. Additionally, it can also be progressed by delving into executing it for multi-lingual and multiple documents conjointly.

REFERENCES

1. arXiv:1704.03242v1 [cs.CL].
2. D. Das and A. F. Martins, "A survey on automatic text summarization," Literature Survey for the Language and Statistics II course at CMU, vol. 4, pp. 192-195, 2007.
3. H. P. Luhn, "The automatic creation of literature abstracts," IBM Journal of research and development, vol. 2, no. 2, pp. 159–165, 1958
4. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.188.3982&rep=rep1&type=pdf
5. https://www.researchgate.net/publication/200042361_TextRank_Bringing_Order_into_Text
6. http://www.ajer.org/papers/v6(08)/ZE0608253260.pdf
7. P. Werbos. Backpropagation through time: what it does and how to do it. Proceedings of IEEE, 1990
8. D. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. Nature, 323(6088):533–536, 1986.
9. https://arxiv.org/pdf/1409.3215v3.pdf
10. M.Chandra, V.Gupta, and S.Paul, "A statistical approach for automatic text summarization by extraction," In Proceeding of the International Conference on Communication Systems and Network Technologies, IEEE, 2011, pp. 268–271.
11. H. P. Luhn, "The automatic creation of literature abstract," IBM Journal of Research and Development, vol. 2(2), 1958, pp. 159–165.
12. M. R. Murthy, J. V. R. Reddy, P. P. Reddy, S. C. Satapathy, "Statistical approach based keyword extraction aid dimensionality reduction," In Proceedings of the International Conference on Information Systems Design and Intelligent Applications. Springer, 2011.
13. D. D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval," In Proceedings of tenth European Conference on Machine Learning, 1998, Springer- Verlag, pp. 4–15.
14. J. D. M. Rennie, L. Shih, J. Teevan, D. R. Karger, "Tackling the poor assumptions of nave Bayes classifiers," In Proceedings of International Conference on Machine Learning, 2003, Pp. 616–623
15. T. Mouratis, S. Kotsiantis, "Increasing the accuracy of discriminative of multinomial Bayesian classifier in text classification," In Proceedings of fourth International Conference on Computer Sciences and Convergence Information Technology, IEEE, 2009. pp. 1246–1251.
16. C. Y. Lin, "Training a selection function for extraction," In Proceedings of the eighth International Conference on Information and Knowledge Management, ACM, 1999, pp. 55–62.
17. K. Knight, D. Marcu, "Statistics-based summarization- step one: Sentence compression," In Proceeding of the Seventeenth National Conference of the American Association for Artificial Intelligence, 2000, pp. 703–710.

18. L. Rabiner, B. Juang, "An introduction to hidden Markov model," Acoustics Speech and Signal Processing Magazine, vol. 3(1), 2003, pp. 4–16.
19. R. Nag, K. H. Wong, F. Fallside, "Script recognition using hidden Markov models," In Proceedings of International Conference on Acoustics Speech and Signal Processing, IEEE, 1986, pp. 2071–2074.
20. C. Zhou, S. Li, "Research of information extraction algorithm based on hidden Markov model," In Proceedings of second International Conference on Information Science and Engineering, IEEE, 2010, pp. 1– 4.
21. M. Osborne, "Using maximum entropy for sentence extraction," In Proceedings of the AssociaCL-02 Workshop on Automatic Summarization, ACL, 2002, pp. 1–8.
22. B. Pang, L. Lee, S.Vaithyanathan, "Thumbs up? Sentiment classification using machine learning technique," In Proceedings of the Association for Computational Linguistics conference on Empirical methods in Natural Language Processing, ACM, 2002, pp. 79–86.
23. H. L. Chieu, H. T. Ng, "A maximum entropy approach information extraction from semi-structured and free text," In Proceedings of the Eighteenth National Conference on Artificial Intelligence, American Association for Artificial Intelligence, 2002, pp. 786–791.
24. R. Malouf, "A comparison of algorithms for maximum entropy parameter estimation," In Proceedings of sixth conference on Natural Language Learning, ACM, 2002, pp. 1–7.
25. P. Yu, J. Xu, G. L. Zhang, Y. C. Chang, F. Seide, "A hidden state maximum entropy model forward confidence estimation," In International Conference on Acoustic, Speech and Signal Processing, IEEE, 2007, pp. 785–788.
26. M. E. Ruiz, P. Srinivasan, "Automatic text categorization using neural networks," In Proceedings of the Eighth American Society for Information Science/ Classification Research, American Society for Information Science, 1997, pp. 59–72
27. T. Jo, "Ntc (neural network categorizer) neural network for text categorization," International Journal of Information Studies, vol. 2(2), 2010.
28. P. M. Ciarelli, E. Oliveira, C. Badue, A. F. De-Souza, "Multi-label text categorization using a probabilistic neural network," International Journal of Computer Information Systems and Industrial Management Applications, vol. 1, 2009, pp. 133–144.
29. I. Guyon, J. Weston, S. Barnhill, V. Vapnik, "Gene selection for cancer classification using support vector machines" mach. learn. Machine Learning, vol. 46(1–3), 2002, pp. 389–422
30. T. Hirao, Y. Sasaki, H. Isozaki, E. Maeda, "Ntt's text summarization system for duc-2002," In Proceedings of the Document Understanding Conference, 2002, pp. 104–107.
31. L. N. Minh, A. Shimazu, H. P. Xuan, B. H. Tu, S. Horiguchi, "Sentence extraction with support vector machine ensemble," In Proceedings of the First World Congress of the International Federation for Systems Research, 2005, pp. 14-17.
32. J. Morris, G. Hirst, "Lexical cohesion computed by thesaural relations as an indicator of the structure of text," Journal of Computational Linguistics, vol. 17(1), ACM, 1999, pp. 21–48.
33. L. Chengcheng, "Automatic text summarization based on rhetorical structure theory," In International Conference on Computer Application and System Modeling, IEEE, 2010, pp. 595–598.
34. S. Brin, L. Page, "The anatomy of a large-scale hyper textual web search engine," In Proceedings of the Seventh International World Wide Web Conference, Computer Networks and ISDN Systems, Elsevier, 1988, pp. 107-117.
35. D. Horowitz, S. D. Kamvar, "Anatomy of a large-scale social search engine," In Nineteenth International Conference on World Wide Web, ACM, 2010, pp. 431–440.
36. M. Efron, "Information search and retrieval in microblogs," Journal of the American Society for Information Science and Technology, vol. 62(6), 2011, pp. 996–1008.
37. T. K. Landauer, P. W. Foltz, D. Laham, "Introduction to latent semantic analysis," Journal of Discourse Processes, vol. 25(2–3), 1998, pp. 259– 284.
38. B. Rosario, "Latent semantic indexing: An overview," Technical Report of INFOSYS 240, University of California, 2000.
39. T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," Journal of Machine Learning, vol. 42(1–2), 2001, pp. 177– 196.
40. M. Simina, C. Barbu, "Meta latent semantic analysis" In IEEE International conference on Systems, Man and Cybernetics, IEEE, 2004, pp. 3720–3724
41. J. T. Chien. "Adaptive Bayesian latent semantic analysis" IEEE Transactions on Audio, Speech, and Language Processing, vol. 16(1), 2008, pp.198–207
42. D. Wang, T. Li, S. Zhu, C. Ding, "Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization," In Proceedings of the 31th Annual International ACM Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval, 2008. ACM, pp. 307–314.
43. J. H. Lee, S. Park, C. M. Ahn, D. Kim, "Automatic generic document summarization based on non-negative matrix factorization" Journal on Information Processing and Management, vol. 45(1), 2009, pp. 20–34.
44. T. G. Kolda, D. P. O'Leary, "Algorithm 805: Computation and uses of the semi-discrete matrix decomposition," Transactions on Mathematical Software, vol. 26(3), 2000, pp. 415–435.
45. V. Snasel, P. Moravec, J. Pokorny, "Using semi-discrete decomposition for topic identification," In Proceedings of the Eighth International Conference on Intelligent Systems Design and Applications, IEEE, 2008, pp. 415–420.
46. D. Corney, D. Albakour, M. Martinez, and S. Moussa, "What do a million news articles look like?" in Proceedings of the First International Workshop on Recent Trends in News Information Retrieval co-located with 38th European Conference on Information Retrieval (ECIR 2016), Padua, Italy, March 20, 2016., 2016, pp. 42–47. [Online]. Available: http://ceur-ws.org/Vol-1568/paper8.pdf
47. B. Hammer, "Nips papers 1987-2016." [Online]. Available: https://www.kaggle.com/benhamner/ nips-papers
48. https://web.stanford.edu/class/cs224n/reports/2749103.pdf
49. https://arxiv.org/abs/1509.00685
50. https://arxiv.org/abs/1512.01712
51. https://arxiv.org/abs/1704.04368
52. https://arxiv.org/abs/1602.06023
53. https://arxiv.org/abs/1704.04368
54. https://arxiv.org/abs/1512.01712
55. https://arxiv.org/abs/1704.04368

BIOGRAPHIES

Meetkumar Patel is currently pursuing an Undergraduate degree in Computer Engineering from Ahmedabad Institute of Technology in affiliation with Gujarat Technological University. He is interested in Machine Learning, Natural Language Processing, Programming Methodology and Computer Security.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 1, January 2018

Adwaita Chokshi is currently pursuing an Undergraduate degree in Computer Engineering from Ahmedabad Institute of Technology in affiliation with Gujarat Technological University. She is interested in Machine Learning, Deep Learning, Natural Language Processing, Artificial Intelligence and Programming Methodology.

Satyadev Vyas is currently Head of Computer Engineering Department at Ahmedabad Institute of Technology in affiliation with Gujarat Technological University. He has a teaching experience of over 14 years and more than 15 years of industrial experience in R&D. His research work primarily concerns Internet of Things (IoT) and Python.

Khushbu Maurya is currently a professor in Computer Engineering Department at Ahmedabad Institute of Technology in affiliation with Gujarat Technological University. She has an experience of over 8 years in teaching field. Her research work primarily concerns Machine Learning and Artificial Intelligence.