CS 577 Project Report

Name & Student ID: 1. Harsh Vora - A20445400

2. Dimple Mehra – A20457225

Semester: Spring 2020 **Course ID:** CS577

Extractive Text Summarization

Abstract:

Text Summarization is the process of obtaining salient information from an authentic text document. In this technique, the extracted information is achieved as a summarized report and conferred as a concise summary to the user; it is challenging for the user to verve through altogether the information accessible on web.

Reduction of text is a very complex problem which, in spite of the progress in the area thus far, poses many challenges to the scientific community. It is also relevant application in today’s information society given the exponential growth of textual information online and the need to promptly assess the contents of text collections. It has long been assumed that summarization presupposes to understand the input text, which means for identifying the important point of the document, explicit representation of the text must be calculated therefore, text summarization became an interesting application to test the understanding capabilities of artificial systems.

1. Problem Statement

Humans accustomed summarize the text by their own, but today thanks to increasing data, it's difficult for the men to cope up with the large data thanks to which the time required for the users to summarize and analyse the large data is increased.
The solution to scale back reading time of the user is producing a succinct document summary.

2. Proposed Solution

Used two layers of Restricted Boltzmann Machine as a Deep Belief Network to reinforce and abstract various features like named entities, proper nouns, numeric tokens, sentence position etc. to attain sentences then selecting the highest scores, hence producing an extractive summary.

3. Implementation details

1. Datasets’s Used:

Dataset we used is “All the news” available on Kaggle which comprises articles from different publications(New York Times, Breitbart, CNN, Business Insider, the Atlantic, Fox News, Talking Points Memo, Buzzfeed News, National Review, New York Post, the Guardian, NPR, Reuters, Vox, and the Washington Post) mainly from the beginning of 2016 to July 2017.

https://www.kaggle.com/snapcrack/all-the-news

2. Data Pre-processing:

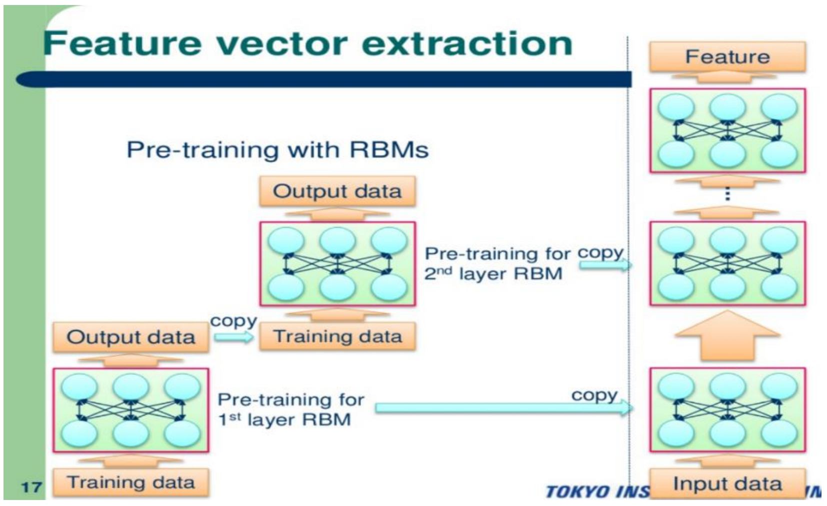
Ambiguities is caused by various verb sorts of one word, different accepted spellings of a specific word, plural and singular terms of the identical things. Moreover, words sort of a, an, the, is, of etc. are referred to as stop words. These are certain high frequency words that don't carry any information and don't serve any purpose towards our goal of summarization. During this phase we do:

1. **Document Segmentation**: The text is split into paragraphs so on keep a track of which paragraph each sentence belongs to and what's the position of a sentence in its respective paragraph.
2. **Paragraph Segmentation**: The paragraphs are further divided into sentences.
3. **Word Normalization**: Each sentence is lessened into words and also the words are normalized. Normalization involves lemmatization and ends up in all words being in one common verb form, crudely stemmed all the way down to their roots with all ambiguities removed. For this purpose, we used Porters algorithm.
4. **Stop Word Filtering**: Each token is analysed to get rid of high frequency stop words.
5. **PoS Tagging**: Remaining tokens are Part-of-Speech tagged into verb, noun, adjective etc. using the PoS Tagging module supplied by NLTK.

3. Feature Extraction:

Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing. A characteristic of these large data sets is a large number of variables that require a lot of computing resources to process.

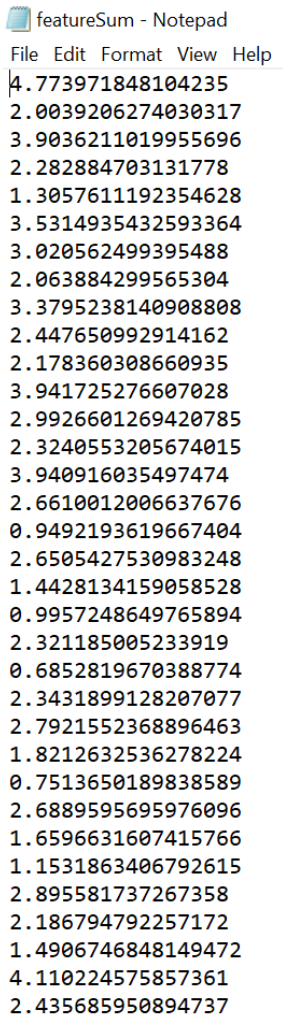
In our model, once the complexity has been reduced and ambiguities are removed, the document is structured into a sentence-feature matrix. A feature vector is extracted for every sentence. These feature vectors frame the matrix. we've experimented with various features. the mix of the subsequent 9 sentence features has clad best suited to summarize factual reports.



These computations are done on the text obtained after the pre-processing phase:

1. **Number of thematic words:** The ten most often occurring words of the text are found. These are thematic words. For every sentence, the ratio of no. of thematic words to total words is calculated.
2. **Sentence position:** This feature is calculated by checking variable SenPos whose value is equal to 1 if the position of text is first or last within the sentence otherwise it's a cosine function of ratio of (SenPos – min) and 1/max and subtracted min from resulting value.
   where, SenPos = position of sentence within the text
   min = threshold(calculated as 0.2 x N) x N
   max = threshold(calculated as 0.2 x N) x 2 x N
   N is total number of sentences in document
   By this, we get a high feature value towards the start and ending of the document, and a progressively decremented value towards the centre.
3. **Sentence length:** This feature is employed to exclude sentences that are too short as those sentences won't be ready to convey much information.
   Variable Sentence_Length is capable 0 if the amount of words is a smaller amount than 3 and otherwise it's capable no. of words within the sentence.
4. **Sentence position relative to paragraph:** This comes directly from the observation that at the beginning of every paragraph, a brand new discussion is begun and at the top of every paragraph, we've a conclusive closing.
   Variable Position_In_Para is capable 1 if it's first or last sentence of a paragraph and otherwise it's capable 0.
5. **Number of proper nouns:** This feature is employed to relinquish importance to sentences having a considerable number of proper nouns. Here, we count the overall number of words that are PoS tagged as proper nouns for every sentence.
6. **Number of numerals:** Since figures are always crucial to presenting facts, this feature gives importance to sentences having certain figures. for every sentence we calculate the ratio of numerals to total number of words within the sentence.
7. **Number of named entities:** Here, we count the overall number of named entities in each sentence. Sentences having references to named entities sort of a company, a gaggle of individuals etc. are often quite important to form any sense of a factual report.
8. **Term Frequency-Inverse Sentence Frequency (TF ISF):** Since we are working with one document, we've taken TF-ISF feature into consideration instead of TF-IDF. Frequency of every word in a very particular sentence is multiplied by the overall number of occurrences of that word all told the opposite sentences. We calculate this product and add it over all words.
9. **Sentence to Centroid similarity:** Sentence having the best TF-ISF score is taken into account because the centroid sentence. Then, we calculate cosine similarity of every sentence thereupon centroid sentence.
   At the top of this phase, we've a sentence-feature matrix.

Output: featureSum matrix :


featureSum - Notepad

File Edit Format View Help

```
4.773971848104235
2.0039206274030317
3.9036211019955696
2.282884703131778
1.3057611192354628
3.5314935432593364
3.020562499395488
2.063884299565304
3.3795238140908808
2.447650992914162
2.178360308660935
3.941725276607028
2.9926601269420785
2.3240553205674015
3.940916035497474
2.6610012006637676
0.9492193619667404
2.6505427530983248
1.4428134159058528
0.9957248649765894
2.321185005233919
0.6852819670388774
2.3431899128207077
2.7921552368896463
1.8212632536278224
0.7513650189838589
2.6889595695976096
1.6596631607415766
1.1531863406792615
2.895581737267358
2.186794792257172
1.4906746848149472
4.110224575857361
2.435685950894737
```

featureSum.txt

Attached is the text file for featureSum matrix:

4. Feature Enhancement:

The sentence-feature matrix has been generated with each sentence having 9 feature vector values. After this, recalculation is completed on this matrix to reinforce and abstract the feature vectors, so on build complex features out of easy ones. This step improves the standard of the summary.

To enhance and abstract, the sentence-feature matrix is given as input to a Restricted Boltzmann Machine (RBM) which has one hidden layer and one visible layer. one hidden layers will suffice for the training process supported the dimensions of our training data.

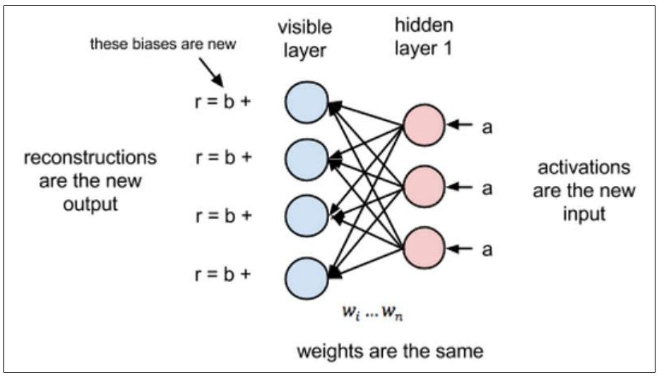
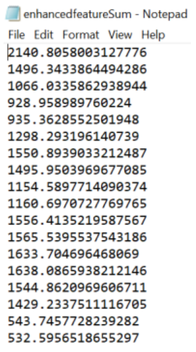


Fig. Restricted Boltzmann Machine (RBM)

The RBM that we are using has 9 perceptrons in each layer with a learning rate of 0.1. We use Persistent Contrastive Divergence method to sample during the training process [17]. we've trained the RBM for five epochs with a batch size of 4 and 4 parallel Gibbs Chains, used for sampling using Persistent CD method. Each sentence feature vector is more established the hidden layer during which feature vector values for every sentence are multiplied by learned weights and a bias value is added to all or any the feature vector values which is additionally learned by the RBM. At the end, we've a refined and enhanced matrix. Note that the RBM will should be trained for every new document that needs to be summarized. the concept is that no document is summarized without going over it. Since each document is exclusive within the features extracted in section 3.2, the RBM will should be freshly trained for every new document.

<u>Output:</u> Enhanced <u>FeatureSum matrix :</u>



```
2140.8058003127776
1496.3433864494286
1066.0335862938944
928.958989760224
935.3628552501948
1298.293196140739
1550.8939033212487
1495.9503969677085
1154.5897714090374
1160.6970727769765
1556.4135219587567
1565.5395537543186
1633.704696468069
1638.0865938212146
1544.8620969606711
1429.2337511116705
543.7457728239282
532.5956518655297
```

Attached is the text file for Enhanced featureSum matrix: enhancedfeatureSum.txt

5. <u>Summary Generation:</u>

The enhanced feature vector values are summed to come up with a score against each sentence. The sentences are then sorted in line with decreasing score value. the foremost relevant sentence is that the first sentence during this sorted list and is chosen as a part of the subset of sentences which can form the summary. Then the following sentence we select is that the sentence having highest Jaccard similarity with the primary sentence, selected strictly from the highest 1/2 the sorted list. This process is recursively and incrementally repeated to pick out more sentences until a user specified summary limit is reached. The sentences are then re-arranged within the order of appearance within the original text. This produces a coherent summary instead of a group of haywire sentences.

4. Results and Discussion:

We have tested the model on several text articles and generated the output(a summary of respective article) and attached are the results for reference:

input-output.pdf

Several factual reports from various domains of health, technology, news, sports etc. with varying number of sentences were used for experimentation and evaluation. The proposed algorithm was run on each of these and system-generated summaries were compared to the summaries produced by humans.

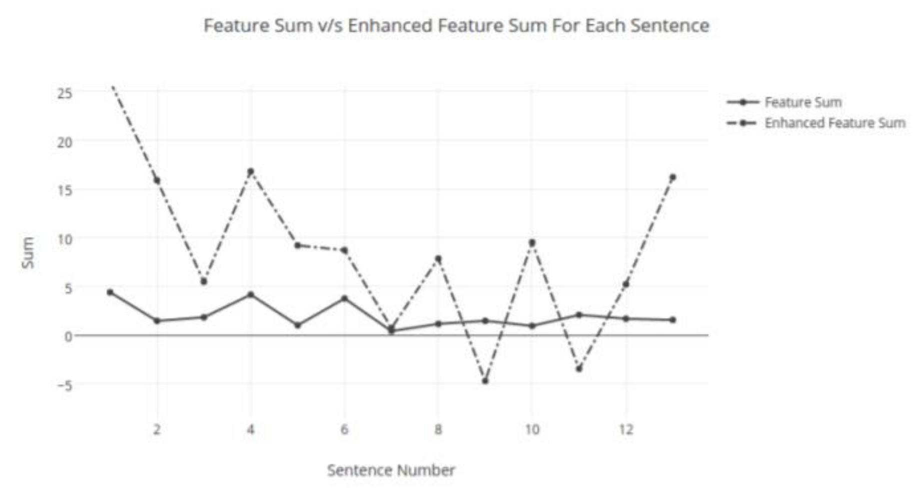


Fig. 2. Comparison between feature vector sum and enhanced feature vector sum

Feature Extraction and Enhancement is disbursed as proposed in above sections for all documents. The values of feature vector and enhanced feature vector for every sentence of 1 such document are plotted in Fig 2. The Restricted Boltzmann Machine has extracted a hierarchical representation out of information that originally didn't have much variation, hence discovering the latent factors. The sentences have then been ranked on the premise of ultimate feature vector and summaries are generated.

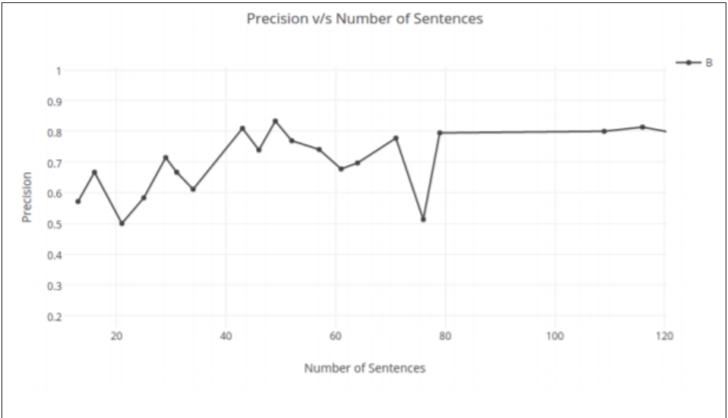


Fig. 3. Precision values corresponding to summaries of various documents

Evaluation of the system-generated summaries is completed supported three basic measures: Precision, Recall and F-Measure [18].

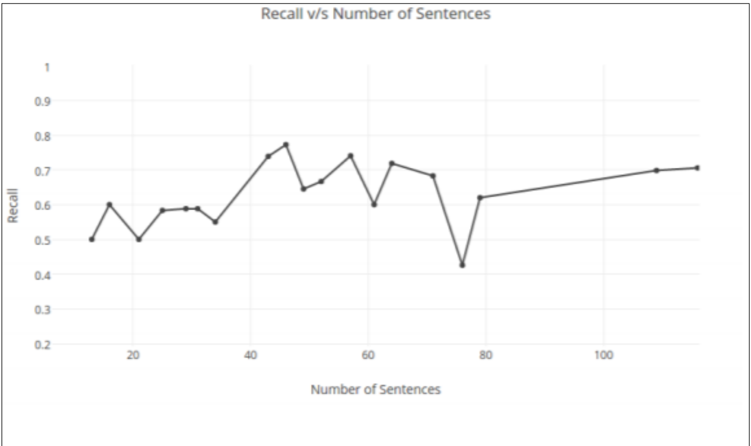


Fig. 4. Recall values corresponding to summaries of various documents

It is seen that because the number of sentences within the original document cross a particular threshold, the Restricted Boltzmann Machine has ample data to be trained successfully and summaries with high precision and recall values are generated. See Fig 3 and 4.

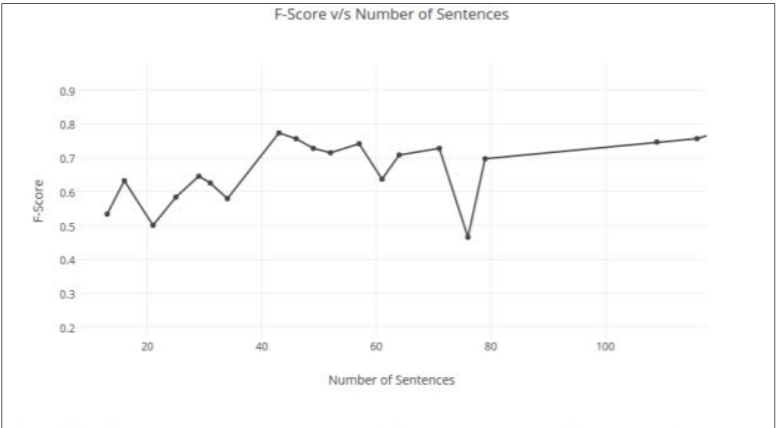
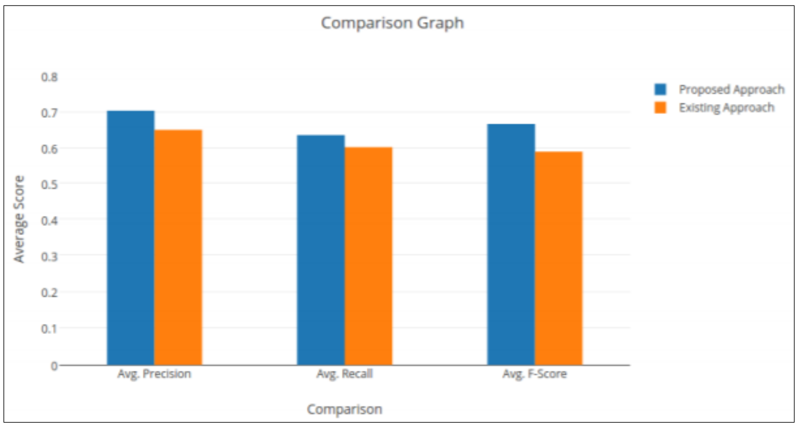


Fig. 5. F-Measure values corresponding to summaries of various documents

Comparative Analysis:

The existing approach was executed for the same set of articles with just one layer of RBM, rather than two as it specifies and average values of Precision, Recall and F-Measure were plotted for drawing a comparison between the existing approach and the proposed approach, while keeping the amount of computation constant.



The proposed approach has an average precision value of 0.7 and average recall value of 0.63 which are both higher than those of the existing approach. Hence, the proposed approach responds better for summarization of factual reports.

Conclusion:

We have developed an algorithm to summarize single-document factual reports. The algorithm runs separately for every input document, rather than learning rules from a corpus, as each document is exclusive in itself. this can be a bonus that our approach provides. We extract 9 features from the given document and enhance them to attain each sentence. Recent approaches are using 2 RBMs stacked on top of every other for feature enhancement. Our approach uses just one RBM and, works effectively and efficiently for factual reports. This has been demonstrated by hand-picking factual descriptions from several domains. This approach can further be developed by adapting the extracted features as per the user's requirements and further adjusting the hyperparameters of the RBM to attenuate processing and error in encoded values.

5. References:

1. **Research Paper-:** Extractive Text Summarization Using Deep Learning
    - **Author's:** Nikhil S. Shirwandkar, Samidha Kulkarni
    - **Year of publication:** 2019
    - **Name of publication:** IEEE (2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA))

    Text Summarization Using Deep Learning.

2. Zhang, Y., Wang, D., Li, T.: iDVS - An Interactive Multi-Document Visual Summarization System: Machine Learning and Knowledge Discovery in Databases, LNCS,
   vol. 6913, pp. 569-584. Springer, Heidelberg (2006). doi: 10.1007/978-3-642-23808-

6 37

3. Wan, X., Xiao, J.: Single Document Keyphrase Extraction using Neighborhood Knowledge. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (2008).
4. H. P. Luhn, “The automatic creation of literature abstracts,” IBM Journal of Research Development, vol. 2, no. 2, pp. 159-165, 1958.
5. RBM, rbm.py [Online] Available: http://deeplearning.net/tutorial/code/rbm.py [Accessed: April 22, 2020]
6. Logistic_sgd, logistic_sgd.py [Online] Available: http://deeplearning.net/tutorial/code/logistic_sgd.py [Accessed: April 22, 2020]