Document summarisation by key sentence extraction using neural networks

RUDRAJIT DAS - 140020012
ADITYA GOLATKAR - 14B030009
AKASH DOSHI - 140010008
DEBARNAB MITRA - 140070037

Introduction:

In today's fast paced world, obtaining a quick summary of lengthy documents is of paramount importance. **Sentence extraction** is a technique used for **automatic summarization of a text**. In this shallow approach, **statistical heuristics** are used to identify the most salient sentences of a text. Sentence extraction is a **low-cost approach** compared to more knowledge-intensive deeper approaches which require additional knowledge bases such as ontologies or linguistic knowledge. In short "sentence extraction" works as a **filter which allows only important sentences to pass**.

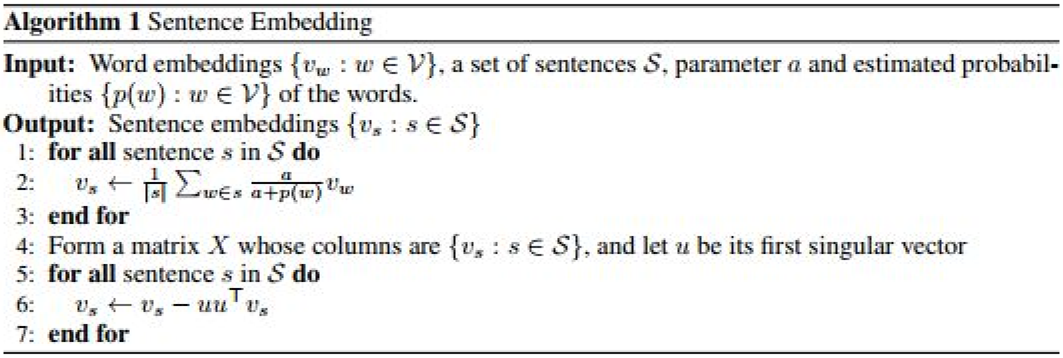
Our aim in this project is to extract key sentences from a document using **supervised learning**. We will explore and compare the performance of two popular forms of neural networks for this task **- feedforward neural networks with ensemble learning** and **convolutional neural networks**.

Sentence Embedding Heuristic:

(Taken from Arora, Sanjeev, Yingyu Liang, and Tengyu Ma. "A simple but tough-to-beat baseline for sentence embeddings." (2016).)

Just compute the **weighted average of the word vectors** in the sentence and then **remove the projections** of the average vectors on their **first principal component** ("common component removal"). Here the weight of a word w is **a/(a + p(w))** with a being a parameter and p(w) the (estimated) word frequency. This method achieves significantly better performance than the unweighted average on a variety of textual similarity tasks, and on most of these tasks even beats some sophisticated supervised methods tested in (Wieting et al., 2016), including some RNN and LSTM models

Most word embedding methods, since they seek to capture word co occurrence probabilities using vector inner product, end up giving large vectors to frequent words, as well as giving unnecessarily large inner products to word pairs, simply to fit the empirical observation that words sometimes occur out of context in documents. These anomalies cause the average of word vectors to have huge components along semantically meaningless directions.



```
Algorithm 1 Sentence Embedding
Input: Word embeddings $\{v_w : w \in \mathcal{V}\}$, a set of sentences $\mathcal{S}$, parameter $a$ and estimated probabilities $\{p(w) : w \in \mathcal{V}\}$ of the words.
Output: Sentence embeddings $\{v_s : s \in \mathcal{S}\}$
for all sentence $s$ in $\mathcal{S}$ do
    $v_s \leftarrow \frac{1}{|s|} \sum_{w \in s} \frac{a}{a + p(w)} v_w$
end for
Form a matrix $X$ whose columns are $\{v_s : s \in \mathcal{S}\}$, and let $u$ be its first singular vector
for all sentence $s$ in $\mathcal{S}$ do
    $v_s \leftarrow v_s - u u^\top v_s$
end for
```

Theory behind sentence embedding:

The latent variable generative model treats corpus generation as a dynamic process, where the tth word is produced at step t. The process is driven by the random walk of a discourse vector c_t.

The discourse vector represents “what is being talked about.” The inner product between the discourse vector c_t and the word vector v_w for word w captures the correlations between them. Then we have $\Pr[w \text{ emitted at time } t \mid c_t] \propto \exp\left(\langle c_t, v_w \rangle\right)$

To achieve a more realistic modelling, the probability of a word w is emitted in the sentence s is modelled by

$$\Pr[w \text{ emitted in sentence } s \mid c_s] = \alpha p(w) + (1 - \alpha) \frac{\exp\left(\langle \tilde{c}_s, v_w \rangle\right)}{Z_{\tilde{c}_s}},$$
$$\text{where } \tilde{c}_s = \beta c_0 + (1 - \beta) c_s, \;\; c_0 \perp c_s$$

Two terms have been introduced here. The first is an additive term **αp(w)** where **p(w)** is the unigram probability (in the entire corpus) of word and **α** is a scalar. This allows words to occur even if their vectors have very low inner products with $\mathbf{c_s}$.

The second is a common discourse vector $\mathbf{c_0}$ which serves as a correction term for the most frequent discourse that is often related to syntax. It boosts the co-occurrence probability of words that have a high component along $\mathbf{c_0}$.

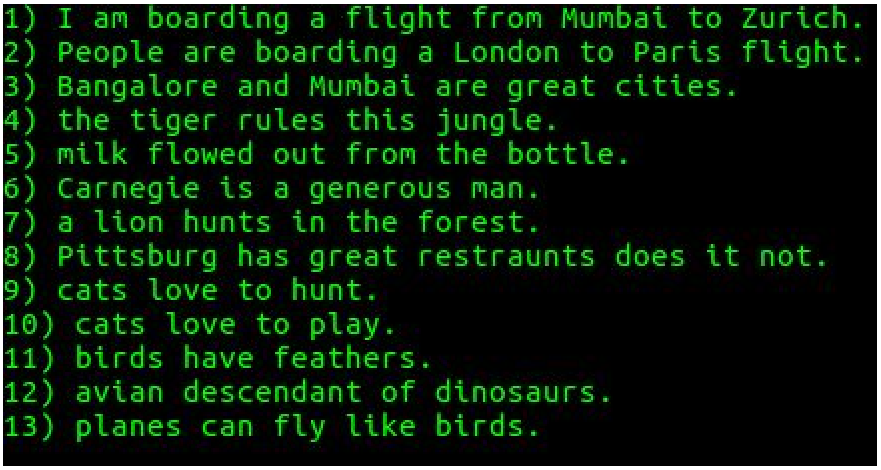
The sentence embedding will be defined as the max likelihood estimate for the vector $\mathbf{c_s}$ that generated it. This can then be shown to generate the coefficients **a/(a + p(w))** where **a = (1 – α)/ αZ**

To estimate $\mathbf{c}_s$, we estimate the direction $\mathbf{c}_0$ by computing the first principal component of $\mathbf{c}^{\sim}_{s}$**’s** for a set of sentences. In other words, the final sentence embedding is obtained by subtracting the projection of $\mathbf{c}^{\sim}_{s}$**’s** to their first principal component.

Feature Extraction:

1. We have considered a corpus of news articles given to us along with their associated summaries. Using the Natural Language Toolkit(NLTK) available in Python, we tokenized each article into its sentences. From each sentence, we then extracted the words
2. We now used the Google Dataset Word2Vec to find the embedding vector for each word. This dataset takes into account all possible similarities between words, for eg. Mumbai and Maharashtra have vector representations which are not too different. It represent words as continuous vectors in a low dimensional space and captures lexical and semantic properties of the words.
3. These word vectors have to be linearly combined to give the sentence vector representation. The coefficients used for the linear combination were described in the sentence embedding heuristic.
4. We now need to label each of the sentences as key or not-key sentence. For this we take each sentence in its summary, and compute its dot product with each sentence in the article. The one with the max dot product is chosen as the key sentence(Cosine Similarity)

Consider the example below for demonstration:



```
1) I am boarding a flight from Mumbai to Zurich.
2) People are boarding a London to Paris flight.
3) Bangalore and Mumbai are great cities.
4) the tiger rules this jungle.
5) milk flowed out from the bottle.
6) Carnegie is a generous man.
7) a lion hunts in the forest.
8) Pittsburg has great restraunts does it not.
9) cats love to hunt.
10) cats love to play.
11) birds have feathers.
12) avian descendant of dinosaurs.
13) planes can fly like birds.
```

Similarity vector for **6)** w/o using the prev. algo: [0.34246296, 0.31892182, 0.22637744, 0.27581882, 0.28438932, 1., **0.39794646**, 0.35118331, 0.24474546, 0.24870716, 0.11177552, 0.20496573, 0.10163028]

Similarity vector for **6)** using the prev. algo : [0.08545813, 0.03463596, 0.04396988, -0.03900207, 0.17517454, 1. , 0.09018801, **0.2086024** , 0.01897737, 0.04286888, -0.23284624, 0.03852853, -0.25666001]

5. Having obtained a label for each sentence in the news article, we proceed to train it, using three different approaches: Fully Connected Neural Network, Convolutional Neural Network and Bagging(Weighted combination of the output of multiple neural networks).

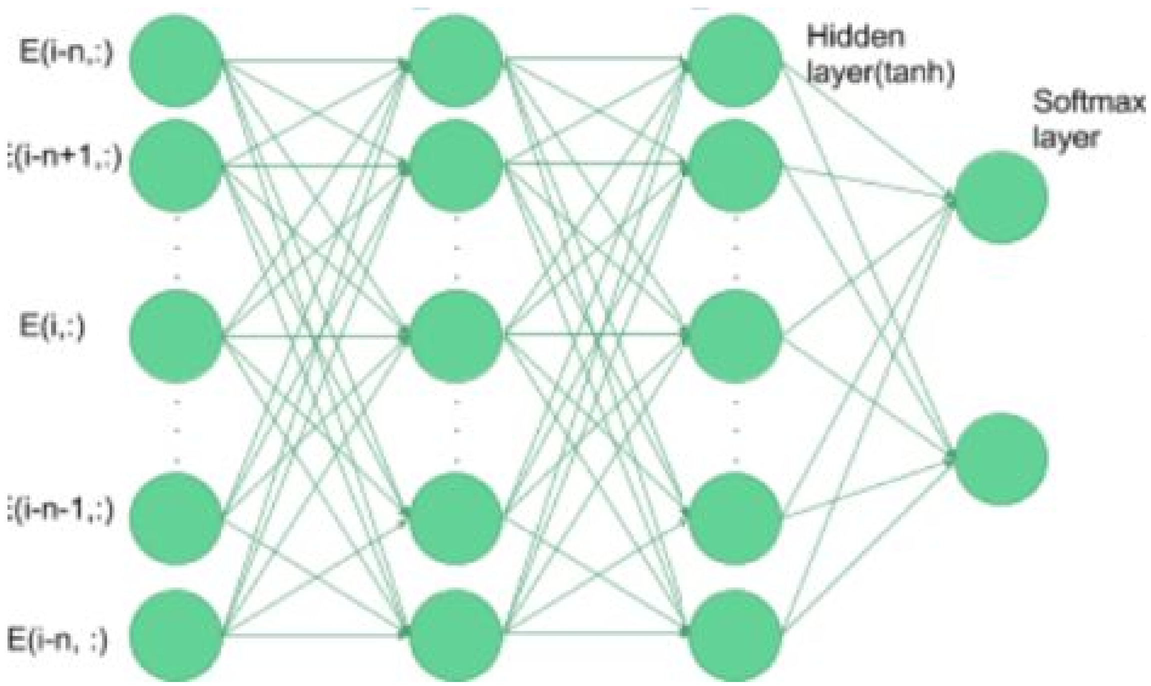
Fully Connected Neural Network:

(Taken from http://cs229.stanford.edu/proj2016/report/tang-KeySentenceExtractionWithNeuralNetwork-report.pdf)

The embedding vectors of each sentence and its context window are fed in one at a time as input to the neural network. e.g. given a sentence S , we consider the surrounding sentences $\{S_{i-n}, S_{i-n+1}...S_{i-1},S_{i+1},..S_{i+n}\}$ as its context window. Also for each sentence S , we have constructed an embedding vector as explained before. For feedforward network, for each sentence, we concat and flatten $\{ E_{i-n}, ..E_{i} ...., E_{i+n}\}$ as $F_i \in R^{(2n+1)m\times 1}$ input feature which is a single column matrix. The label of the central sentence in the window is used as the training label for that window. So what we are predicting is $P(S_i$ is key sentence$| C_i)$ where S_i is the context window.

To avoid biasing the neural network to the not-key outcome(which is the vast majority), we are roughly training double the number of negative sentences as positive sentences. For this, we have randomly sampled from our overall dataset.

The neural network contains two **tanh** layers and a final **sigmoid** layer. However, the data has not been normalized as that was found to adversely affect the performance.



Ensemble Learning of Neural Networks:

(Taken from Zhou, Zhi-Hua, Jianxin Wu, and Wei Tang. "Ensembling neural networks: many could be better than all." *Artificial intelligence* 137.1-2 (2002): 239-263.)

In this paper, the relationship between the ensemble and its component neural networks is analyzed from the context of both regression and classification, which reveals that it may be better to ensemble **many** instead of **all** of the neural networks at hand.

Since using a single neural network did not perform very well, we tried using ensemble learning on NNs, specifically bagging. Each NN in the ensemble is trained using a randomly drawn subset of the training set (with the ratio of negative to positive examples being nearly 2:1), and the average of the results of a **subset of these NNs** in the ensemble is taken as the final result. Seeking inspiration from the above mentioned paper, we choose those NN models for averaging whose validation accuracy exceeds a preset threshold λ. The paper suggests including those NN's whose weight exceeds a threshold λ, assuming that each neural network can be assigned a weight that could characterize the fitness of including this network in the ensemble. An objective function is defined which is basically the inverse of the cross-validation error. The set of weights which maximize this objective function are our optimal weights (optimization performed by using genetic algorithms). In order to avoid all this hassle, we consider only those NN's which have an accuracy of > **0.5** (after some tuning) for averaging.

Convolutional Neural Network:

(Taken from Yoon Kim, Convolutional Neural Networks for Sentence Classification, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751, October 2529, 2014, Doha, Qatar**)**

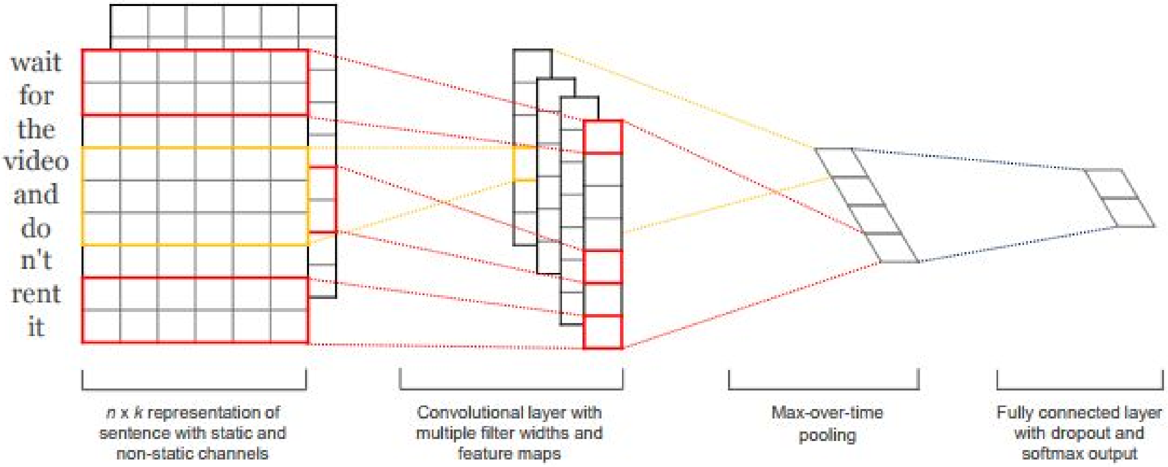

wait
for
the
video
and
do
n't
rent
it
n x *k* representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 1: Model architecture with two channels for an example sentence.

- The model architecture, shown in Figure 1, is a slight variant of the CNN architecture of Collobert et al. (2011). Let $x_i \in R^k$ be the k-dimensional sentence vector corresponding to the i-th word in the article. A paragraph of length n (padded where necessary) is represented as $\mathbf{x_{1:n}} = \mathbf{x_1} \oplus \mathbf{x_2} \oplus \mathbf{. . .} \oplus \mathbf{x_n}$, where $\oplus$ is the concatenation operator. In general, let $x_{i:i+j}$ refer to the concatenation of sentences x_i , x_{i+1}, . . . , x_{i+j} . A convolution operation involves a filter $w \in R^{hk}$, which is applied to a window of h words to produce a new feature.
- For example, a feature c_i is generated from a window of words $x_{i:i+h-1}$ by $c_i = f(w \cdot x_{i:i+h-1} + b)$. Here $b \in R$ is a bias term and f is a nonlinear function such as the hyperbolic tangent. This filter is applied to each possible window of words in the sentence $\{x_{1:h}, x_{2:h+1}, . . . , x_{n-h+1:n}\}$ to produce a feature map $c = [c_1, c_2, . . . , c_{n-h+1}]$, (3) with $c \in R^{\,n-h+1}$.
- We then apply a max-overtime pooling operation (Collobert et al., 2011) over the feature map and take the maximum value $c\hat{} = \max\{c\}$ as the feature corresponding to this particular filter.
- The idea is to capture the most important feature—one with the highest value—for each feature map. This pooling scheme naturally deals with variable sentence lengths. We have described the process by which one feature is extracted from one filter. The model uses multiple filters (with varying window sizes) to obtain

multiple features. These features form the penultimate layer and are passed to a fully connected softmax layer whose output is the probability distribution over labels

Convolutional Neural Networks Modification:

The CNN architecture described above did not yield very promising results. Instead, we treated the 300x7 matrix as an single channel image. We then used a convolutional layer with 128 filters and 3x3 kernel. We used “tanh” activation function. Next, we added a 2x2 pooling layer. We then flattened the output of the max-pooling layer and connected it to a 256x1 fully connected layer and then finally to a sigmoid neuron. For regularization (to prevent overfitting) we incorporated a dropout of 0.5. Using this modified architecture, we got better results.

Results achieved:

1. Fully Connected Neural Network

Accuracy = 0.65

F1 score = 0.29

Some examples:

1.

Actual summary:

Setting aside all talks of alleged cold war, Captain Amarinder Singh and cricketer-turned-politician Navjot Singh Sidhu, who recently joined the Congress, held a joint press conference in Amritsar today.To this Captain said, "I am his wicketkeeper.I will catch him in the slips.

Obtained summary:

Cricketer-turned-politician Navjot Singh Sidhu, who recently joined the Congress, held a joint press conference with Captain Amarinder Singh on Thursday and said, "Baap baap hota hai, beta beta hota hai." Dismissing rumours that Sidhu is not on agreeable terms with Amarinder Singh, the Captain said, "I am his wicketkeeper. I will catch him in the slips."

2.

Actual summary:

The Maharashtra Navnirman Sena (MNS) has said that it will continue to oppose Indian films featuring Pakistani actors. Ameya Khopkar, who heads MNS' cinema wing, said that even though Pakistan has resumed screening Indian films, the party's stand will remain unchanged, until Pakistan stops attacks on Indian lands. Notably, Pakistani actress Mahira Khan stars in Shah Rukh Khan's film 'Raees'.

Note: Third line in the Actual summary is not there in the document.

Obtained NN summary:
Mumbai, Dec 19 (PTI Even as cinemas in Pakistan today began screening Indian movies, over two months after film exhibitors and theatre owners suspended it amidst Indo-Pak tensions, the Maharashtra Navnirmam Sena (MNS) said it will continue to oppose Indian films featuring Pakistani actors."Though Pakistan resumes screening of Indian cinema, our stand will remain unchanged, until Pak stops attacks on Indian land," Ameya Khopkar, head of the film wing of Raj Thackeray-led MNS, said in a tweet today.

3.
Actual summary:
Google India has partnered with the Ministry of Consumer Affairs for a campaign called 'Digitally Safe Consumer' to help protect consumer interest online. The company has said it will provide training material to over 1,200 consumer organisations and consumer affairs departments of all Indian states. Google is also working with schools to spread awareness about internet safety among the youth.

Note: Third line in the Actual summary is not there in the document.

Obtained NN summary:
Global search engine giant Google has tied up with the Ministry of Consumer Affairs to raise awareness about online safety in India.Together, they will embark on a nationwide 'Digitally Safe Consumer' campaign, to help better protect consumer interest online, Google India announced on Saturday.Through the workshops, Google aims to provide people in India the desired training and necessary information on online safety tools.

2. Convolutional Neural Network
Accuracy = 0.78
F1 score = 0.46

Some examples:

1.

Obtained CNN Summary:

Canadian-born Indian YouTube sensation Lilly Singh has emerged as the Favourite YouTube Star 2017 at the 43rd People's Choice Awards!Lilly, who also goes by the name IISuperWomanII, was up against some stiff competition, with YouTubers PewDiePie, Shane Dawson, Tyler Oakley, and Miranda Sings, but walked away as 2017's favourite star.twitter.

Actual Summary:

Indo-Canadian YouTube personality Lilly Singh, also known as Superwoman, was named the Favourite YouTube Star at the 2017 People's Choice Awards in Los Angeles. PewDiePie, Miranda Sings, Shane Dawson and Tyler Oakley were the other nominees in the category. With over 10 million subscribers on YouTube, numerous celebrities including Priyanka Chopra have featured in Lilly's videos.nn

Note: Third line in the Actual summary is not there in the document.

2.

Obtained CNN Summary:

England pace bowler Jake Ball on Tuesday said the visitors will try to unsettle Indian skipper Virat Kohli with short balls and not let him find his rhythm while batting during the second One-Day International (ODI) cricket match.Kohli scored a gritty 122 for the hosts to better the visitors in the opening game of the three-match ODI series at Pune on Sunday, pulling them 1-0 ahead.The second match will be played here on January 19.

Actual Summary:

England fast bowler Jake Ball on Tuesday said that England bowlers will use short balls to unsettle Virat Kohli and will not let him find his rhythm while batting during the second ODI. "He's an unbelievable player. We've got plans for him and, hopefully, we can put them into practice in a couple of days' time," added Ball.

Note: **Better Summary obtained by our system!**

3.

Obtained CNN Summary:

Demonetisation was never an electoral or political decision, it was taken in the long-term interest of our economy, BJP national general secretary Ram Madhav has said.In an interview with India Today, Madhav said, "It's not a matter of winning or losing elections.The decision was taken in the interest of the country and the economy.""Black money is the result of aberrations in the society and whoever is holding the same needs to be punished," he said.

Actual Summary:
BJP National General Secretary Ram Madhav has said demonetisation was never an electoral or political decision, it was taken in the long-term interest of the economy. He further asserted that there was no parallel economy running after demonetisation. "Black money is the result of aberrations in the society and whoever is holding the same needs to be punished," he added.n

4.
Obtained CNN Summary:
Superstar Rajinikanth on Friday extended his support for jallikattu, the popular and ancient bull-taming sport, played usually around Pongal festival in Tamil Nadu.He said it must be held as it is part of Tamil culture.Last year, the Supreme Court banned jallikattu, earning the wrath of its supporters and well-wishers."After Kamal Haasan's statement, several Kollywood celebs backed jallikattu, including Dhanush, Simbu, Khushbu Sundar, GV Prakash and RJ Balaji.

Actual Summary:
Superstar Rajinikanth on Friday extended his support for Jallikattu, the popular and ancient bull-taming sport, played during Pongal festival in Tamil Nadu. Rajinikanth said, "Bring in whatever rules but Jallikattu must be held to keep up the traditions of our Tamil culture." Earlier, Kamal Haasan had also supported Jallikattu, saying, "If you want a ban on Jallikattu, ban biriyani too."

3.
Cases where CNN performed better than Neural Networks-
Obtained CNN Summary:
Canadian-born Indian YouTube sensation Lilly Singh has emerged as the Favourite YouTube Star 2017 at the 43rd People's Choice Awards!Lilly, who also goes by the name IISuperWomanII, was up against some stiff competition, with YouTubers PewDiePie, Shane Dawson, Tyler Oakley, and Miranda Sings, but walked away as 2017's favourite star.twitter.

Obtained NN summary:
Canadian-born Indian YouTube sensation Lilly Singh has emerged as the Favourite YouTube Star 2017 at the 43rd People's Choice Awards!"I am so grateful.

Code Implementation Details:

1. For the Feedforward Neural Net, we used Scikit library functions in Python.
2. We used Gensim for loading Google's word2vec.
3. For the Convolutional Neural Network, we used Keras.

Problems faced:

1. Finding feasible datasets were hard. Moreover there were lot of a spurious characters in the dataset, which caused difficulty in extracting the correct sentences.
2. More importantly, the computation power at our disposal was a severely limiting factor. We were unable to train more than 1500 documents, and this severely affected the accuracy on test articles while identifying the correct sentences.
3. There were many words in the Indian News Articles which did not have a word vector representation in the Google dataset. This affected the key sentences identified.
4. Some of the datasets were difficult to calculate truncated SVD for due to various numerical errors.

Future work:

1. What we have done here is simply to extract the key sentences, we have not formed new sentences in the summary. However by doing a subject-object-predicate analysis, we can construct a dependency graph and each of the edges can be appropriately weighted to determine which predicate-object pairs should be picked for the given subject. This or some other form of abstractive text summarization can be employed.
2. We can improve the sentence embedding algorithm slightly by considering Taylor Series Expansion up to the second order term as it doesn't seem small enough to be neglected. We might also consider approximating the exponential term by a linear term as the inner product could be small enough to perform linear approximation (just to simplify things).

Links to Datasets:

1. https://github.com/tapilab/is-karthikbmk/tree/master/data/DUC2001/Summaries
2. https://www.kaggle.com/sunnysai12345/news-summary
3.https://www.quora.com/What-are-some-interesting-techniques-for-learning-sentence-embeddings