PROJECT REPORT

(Project Term January-May 2021)

TEXT SUMMARIZATION

Submitted by

Name of Student1 : SAIPAVAN SARAKANAM
Registration Number :11904618
Name of Student2 : N. SURYA Kalyan Reddy
Registration Number: 11904411

Course Code : INT246

Under the Guidance of

(Name of faculty mentor with designation) :Dr .Sagar Pande

School of Computer Science and Engineering

LOVELY
PROFESSIONAL
UNIVERSITY

DECLARATION

We hereby declare that the project work entitled Text Summarization is an authentic record of our own work carried out as requirements of Project for the award of B.Tech degree in Computer Science Engineering from Lovely Professional University, Phagwara, under the guidance of Dr . Sagar Pande , during August to November 2020. All the information furnished in this project report is based on our own intensive work and is genuine.

Name of Student 1: SAIPAVAN SARAKANAM
Registration Number: 11904618

Name of Student 2: N. SURYA Kalyan Reddy

Registration Number: 11904411

CERTIFICATE

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Project under my guidance and supervision. The present work is the result of their original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Project is fit for the submission and partial fulfillment of the conditions for the award of B.Tech degree in Computer Science Engineering from Lovely Professional University, Phagwara.

Signature and Name of the Mentor : Dr. Sagar Pande

Designation

School of Computer Science and
Engineering, Lovely Professional University,
Phagwara, Punjab.

Date : 23-11-2021

Text Summarization: An Overview

1.Abstract:

In this new era where tremondous information is available on the internet it is most important to provide the improved mechanism to extract the information quickly and most efficiently It is very difficult for human beings to manually extract the summary of a large documents of text. There are plenty of text material available on the internet. So there is a problem of searching for relevant documents from the number of documents available, and absorbing relevant information from it In order to solve the above two problems, the automatic text summarization is very much necessary Text summarization is the process of identifying the most important meaningful information in a document or set of related documents and compressing them into a shorter version preserving its overall meanings.

2.Introduction:

Before going to the Text summarization, first we, have to know that what a summary is. A summary is a text that is produced from one or more texts, that conveys important information in the original text, and it is of a shorter form. The goal of automatic text summarization is presenting the source text into a shorter version with semantics The most important advantage of using a summary is it reduces the reading time. Text Summarization methods can be classified into extractive and abstractive summarization. An extractive summarization method consists of selecting important sentences, paragraphs etc. from the original document and concatenating them into shorter form. An Abstractive summarization is an understanding of the main concepts in a document and then express those concepts in clear natural language.

There are two different groups of text summarization indicative and informative Inductive summarization only represent the main idea of the text to the user. The typical length of this type of summarization is 5 to 10 percent of the main text On the other hand, the informative summarization systems gives concise information of the main text The length of informative summary is 20 to 30 percent of the main text

3.Main steps for text summarization:

There are three main steps for summarizing documents. These are topic identification, interpretation and summary generation.

3.1. Topic Identification The most prominent information in the text is identified There are different techniques for topic identification are used which are Position, Cue Phrases, word frequency Methods which are based on the position of phrases are the most useful methods for topic identification.

3.2. Interpretation Abstract summaries need to go through interpretation step. In This step, different subjects are fused in order to form a general content.

3.3. Summary Generation In this step, the system uses text generation method.

4. Extractive text summarization: This process can be divided into two step : Pre Processing step and Processing step. PreProcessing is structured representation of the original text. It usually includes: a) Sentences boundary identification. In English, sentence boundary is identified with presence of dot at the end of sentence. b) Stop-Word Elimination—Common words with no semantics c) Stemming—The purpose of stemming is to obtain the stem or radix of each word, which emphasize its semantics.

In Processing step, features influencing the relevance of sentences are decided and calculated and then weights are assigned to these features using weight learning method. Final score of each sentence is determined using Feature-weight equation. Top ranked sentences are selected for final summary.

Summary evaluation is a very important aspect for text summarization. Generally, summaries can be evaluated using intrinsic or extrinsic measures. While intrinsic methods attempt to measure summary quality using human evaluation and extrinsic methods measure the same through a task-based performance measure such the information retrieval-oriented task.

5. TEXT SUMMARIZATION HISTORY:

In the past, extractive summarizers have been mostly based on scoring sentences in the source document. The most common and recent text summarization techniques use either statistical approaches, or linguistic techniques The high frequency words standard keyword Cue Method Title Method Location Method are used for weighting the sentences.

6. FEATURES FOR EXTRACTIVE TEXT SUMMARIZATION:

Most of the current automated text summarization systems use extraction method to produce a summary Sentenceextraction techniques are commonly used to produce extraction summaries. One of the methods to obtain suitable sentences is to assign some numerical measure of a sentence for the summary called sentence scoring and then select the best sentences to form document summary based on the compression rate. In the extraction method, compression rate is an important factor used to define the ratio between the length of the summary and the source text. As the compression rate increases, the summary will be larger, and more insignificant content is contained. While the compression rate decreases the summary to be short, more information is lost. In fact, when the compression rate is 5-30%, the quality of summary is acceptable.

7. EXTRACTIVE SUMMARIZATION METHODS :

A. Term Frequency-Inverse Document Frequency (TF-IDF) method:

TF-IDF stands for **“Term Frequency — Inverse Document Frequency”**. This is a technique to quantify words in a set of documents. We generally compute a score for each word to signify its importance in the document and corpus. This method is a widely used technique in Information Retrieval and Text Mining.

If I give you a sentence for example “This building is so tall”. It's easy for us to understand the sentence as we know the semantics of the words and the sentence. But how can any program (eg: python) interpret this sentence? It is easier for any programming language to understand textual data in the form of numerical value. So, for this reason, we need to vectorize all of the text so that it is better represented.

By vectorizing the documents we can further perform multiple tasks such as finding the relevant documents, ranking, clustering, etc. This exact technique is used when you perform a google search (now they are updated to newer transformer techniques). The web pages are called documents and the search text with which you search is called a query. The search engine maintains a fixed representation of all the documents. When you search with a query, the search engine will find the relevance of the query with all of the documents, ranks them in the order of relevance and shows you the top k documents. All of this process is done using the vectorized form of query and documents.

Now coming back to our TF-IDF,

TF-IDF = Term Frequency (TF) * Inverse Document Frequency (IDF)

Terminology

- t — term (word)
- d — document (set of words)
- N — count of corpus
- corpus — the total document set

Term Frequency

This measures the frequency of a word in a document. This highly depends on the length of the document and the generality of the word, for example, a very common word such as “was” can appear multiple times in a document. But if we take two documents with 100 words and 10,000 words respectively, there is a high probability that the common word “was” is present more in the 10,000 worded document. But we cannot say that the longer document is more important than the shorter document. For this exact reason, we perform normalization on the frequency value, we divide the frequency with the total number of words in the document.

Recall that we need to finally vectorize the document. When we plan to vectorize documents, we cannot just consider the words that are present in that particular document. If we do that, then the vector length will be different for both the documents, and it will not be feasible to compute the similarity. So, what we do is that we vectorize the documents on the **vocab**. Vocab is the list of all possible worlds in the corpus.

B. Cluster based method:

1. Overview

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups. In simple words, the aim is to segregate groups with similar traits and assign them into clusters.

Let's understand this with an example. Suppose, you are the head of a rental store and wish to understand preferences of your costumers to scale up your business. Is it possible for you to look at details of each costumer and devise a unique business strategy for each one of them? Definitely not. But, what you can do is to cluster all of your costumers into say 10 groups based on their purchasing habits and use a separate strategy for costumers in each of these 10 groups. And this is what we call clustering.

Now, that we understand what is clustering. Let's take a look at the types of clustering.

2. Types of Clustering

Broadly speaking, clustering can be divided into two subgroups :

- **Hard Clustering:** In hard clustering, each data point either belongs to a cluster completely or not. For example, in the above example each customer is put into one group out of the 10 groups.
- **Soft Clustering**: In soft clustering, instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned. For example, from the above scenario each costumer is assigned a probability to be in either of 10 clusters of the retail store.

3. Types of clustering algorithms

Since the task of clustering is subjective, the means that can be used for achieving this goal are plenty. Every methodology follows a different set of rules for defining the '*similarity*' among data points. In fact, there are more than 100 clustering algorithms known. But few of the algorithms are used popularly, let's look at them in detail:

- **Connectivity models:** As the name suggests, these models are based on the notion that the data points closer in data space exhibit more similarity to each other than the data points lying farther away. These models can follow two approaches. In the first approach, they start with classifying all data points into separate clusters & then aggregating them as the distance decreases. In the second approach, all data points are classified as a single cluster and then partitioned as the distance increases. Also, the choice of distance function is subjective. These models are very easy to interpret but lacks scalability for handling big datasets. Examples of these models are hierarchical clustering algorithm and its variants.

- **Centroid models:** These are iterative clustering algorithms in which the notion of similarity is derived by the closeness of a data point to the centroid of the clusters. K-Means clustering algorithm is a popular algorithm that falls into this category. In these models, the no. of clusters required at the end have to be mentioned beforehand, which makes it important to have prior knowledge of the dataset. These models run iteratively to find the local optima.

- **Distribution models:** These clustering models are based on the notion of how probable is it that all data points in the cluster belong to the same distribution (For example: Normal, Gaussian). These models often suffer from overfitting. A popular example of these models is Expectation-maximization algorithm which uses multivariate normal distributions.

- **Density Models:** These models search the data space for areas of varied density of data points in the data space. It isolates various different density regions and assign the data points within these regions in the same cluster. Popular examples of density models are DBSCAN and OPTICS.

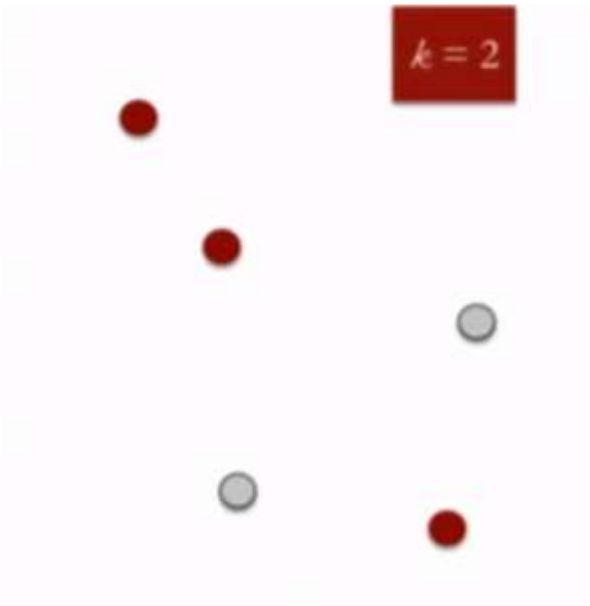
Now I will be taking you through two of the most popular clustering algorithms in detail – K Means clustering and Hierarchical clustering. Let's begin.

4. K Means Clustering

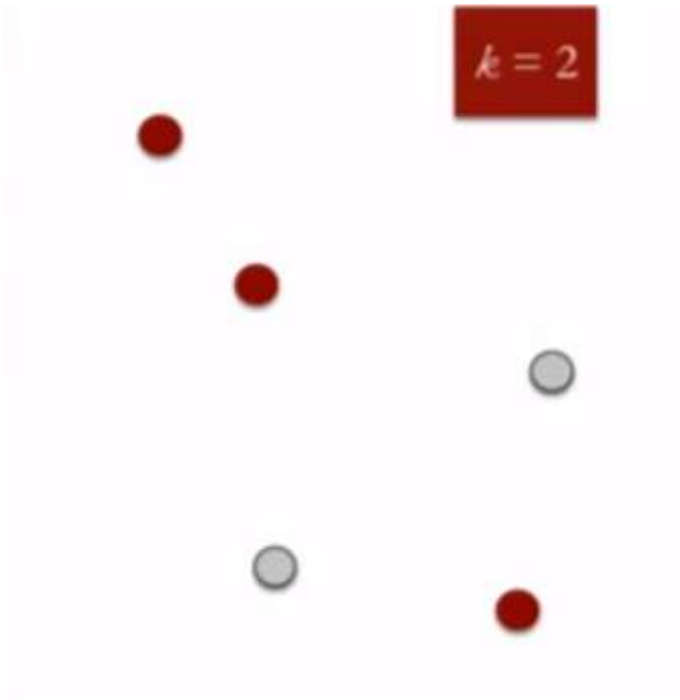
K means is an iterative clustering algorithm that aims to find local maxima in each iteration.

This algorithm works in these 5 steps :

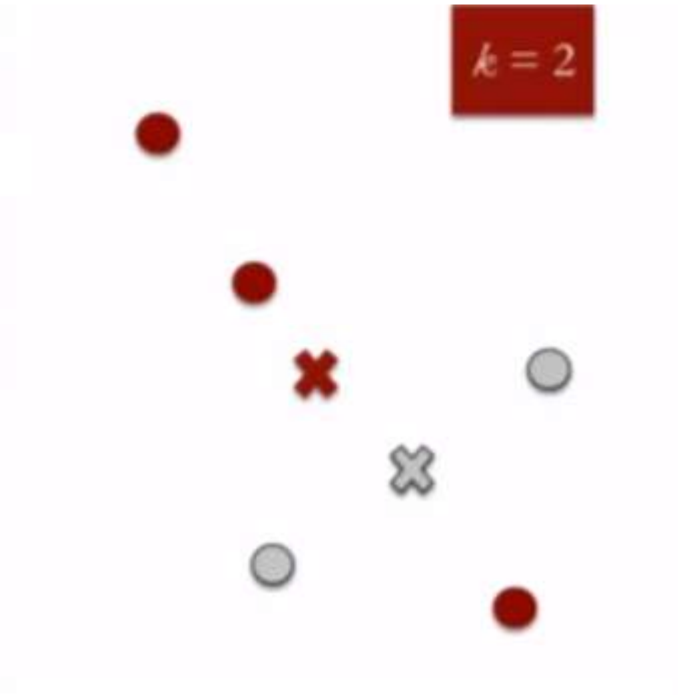
1. Specify the desired number of clusters K : Let us choose k=2 for these 5 data points in 2-D space.



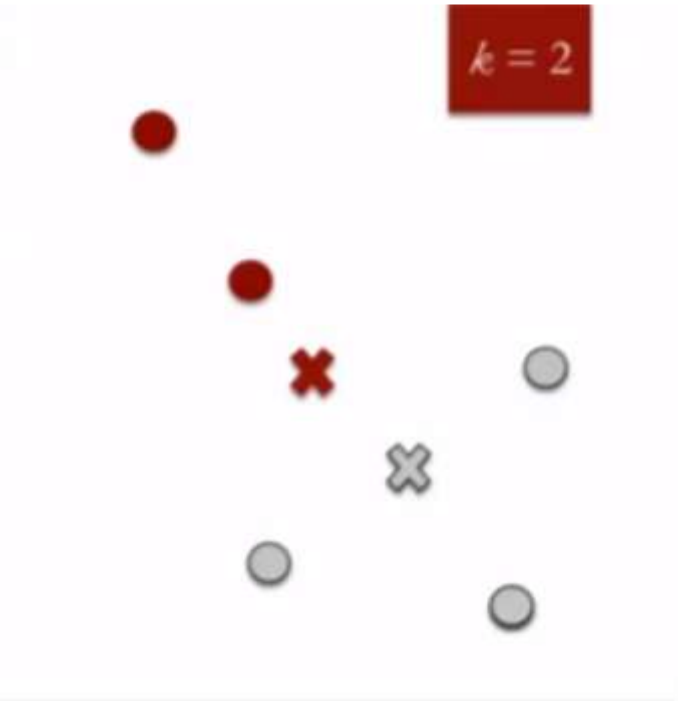
2. Randomly assign each data point to a cluster : Let's assign three points in cluster 1 shown using red color and two points in cluster 2 shown using grey color.



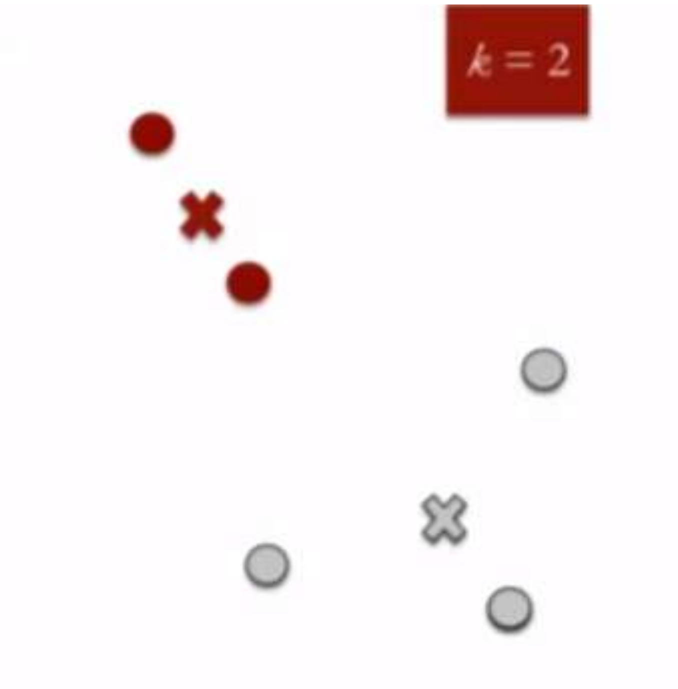
3. Compute cluster centroids : The centroid of data points in the red cluster is shown using red cross and those in grey cluster using grey cross.



4. Re-assign each point to the closest cluster centroid : Note that only the data point at the bottom is assigned to the red cluster even though its closer to the centroid of grey cluster. Thus, we assign that data point into grey cluster

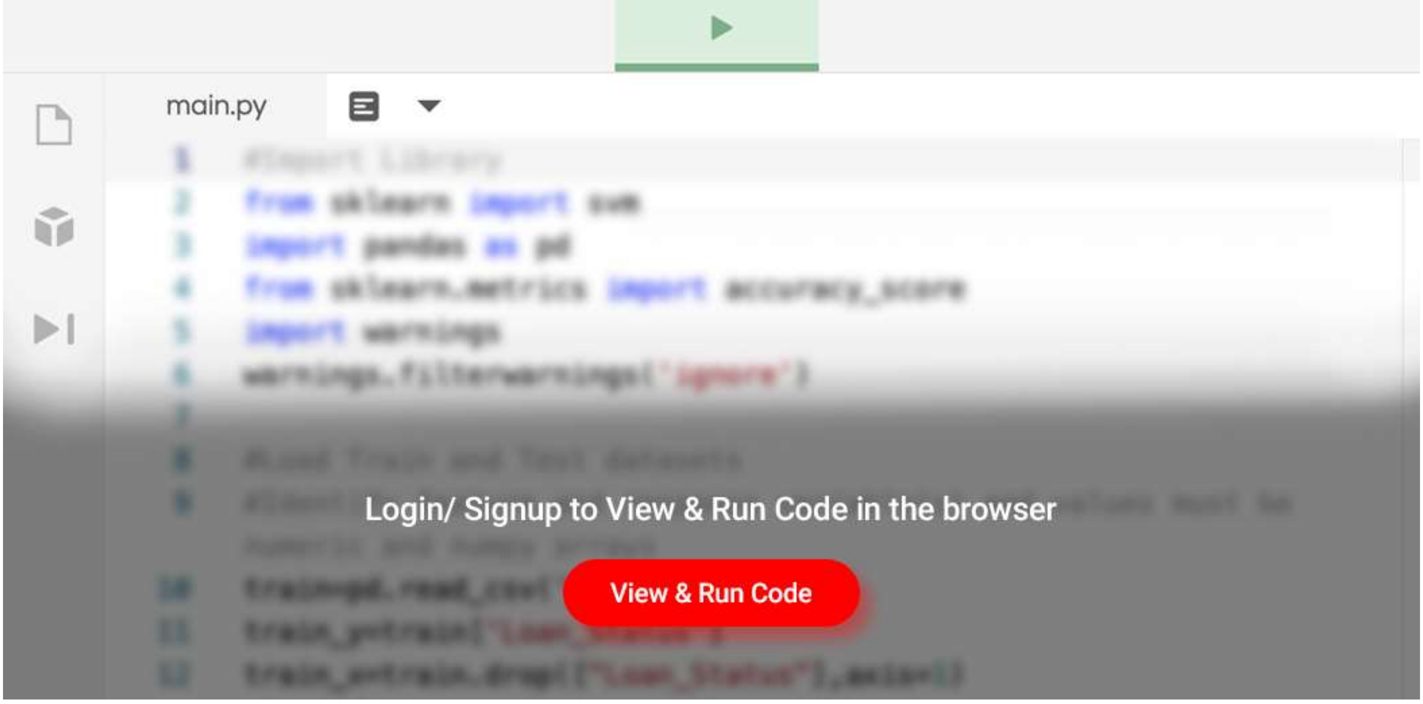


5. Re-compute cluster centroids : Now, re-computing the centroids for both the clusters.



6. Repeat steps 4 and 5 until no improvements are possible : Similarly, we'll repeat the 4th and 5th steps until we'll reach global optima. When there will be no further switching of data points between two clusters for two successive repeats. It will mark the termination of the algorithm if not explicitly mentioned.

Here is a live coding window where you can try out K Means Algorithm using scikit-learn library.



```
#Import Library
from sklearn import svm
import pandas as pd
from sklearn.metrics import accuracy_score
import warnings
warnings.filterwarnings('ignore')

#Load Train and Test datasets
#Identi[illegible] values must be
numeric and numpy arrays
train=pd.read_csv([illegible]
train_y=train['Loan_Status'[illegible]
train_x=train.drop(["Loan_Status"],axis=1)
```

5. Hierarchical Clustering

Hierarchical clustering, as the name suggests is an algorithm that builds hierarchy of clusters. This algorithm starts with all the data points assigned to a cluster of their own. Then two nearest clusters are merged into the same cluster. In the end, this algorithm terminates when there is only a single cluster left.

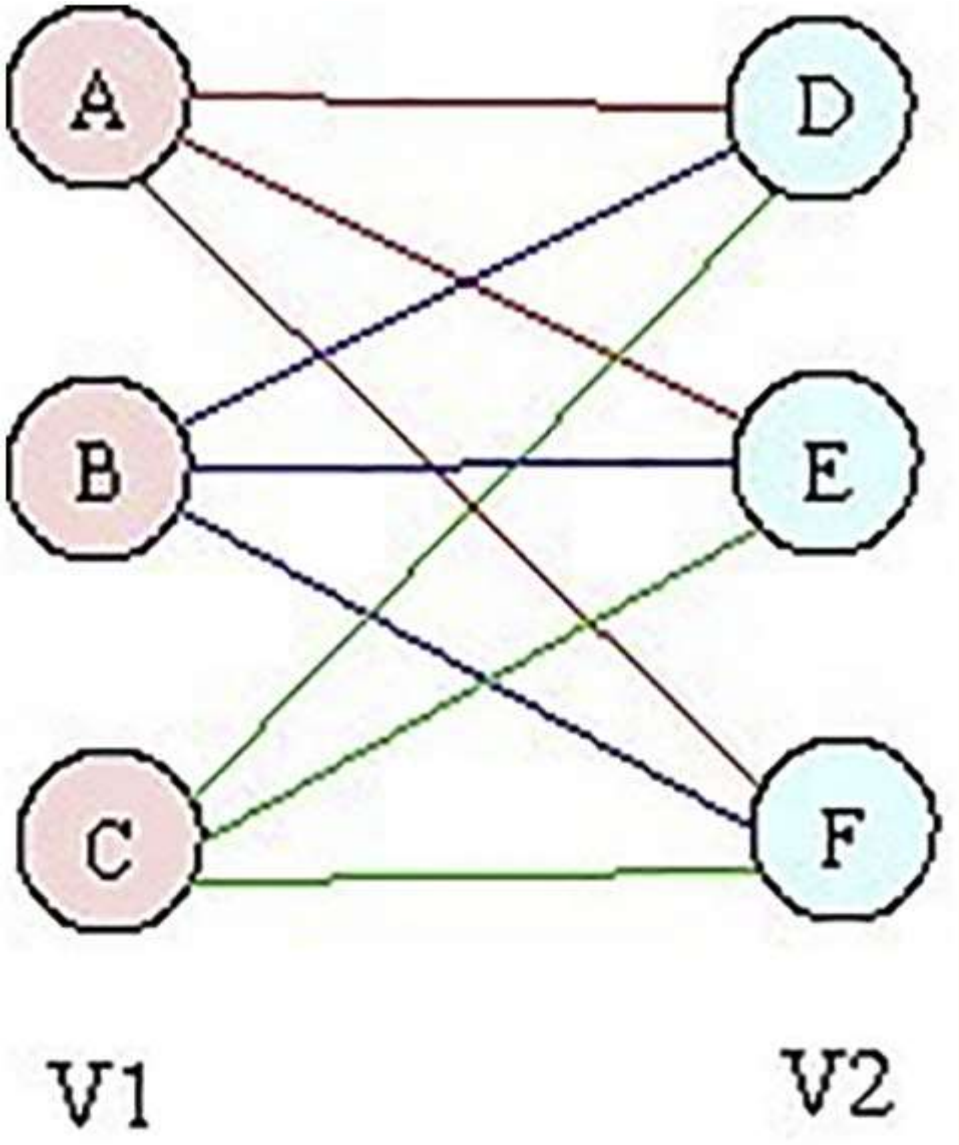
The results of hierarchical clustering can be shown using dendrogram. The dendrogram can be interpreted as:

C. Graph theoretic approach:

D. The graph-theoretic approach is one of the convenient ways of analyzing genome sequences. In our work, we use a complete bipartite graph to represent genome sequences. A complete graph is one where any two vertices are connected by an edge. In particular, a bipartite graph has some special features. Such a graph has two independent sets of graph vertices and that no two graph vertices of the same set are adjacent. A bipartite graph [37] G is denoted by the pair *(V, K)*, where $V = (V_1 , V_2)$ are the two sets of vertices and *K* represents the edges of the graph.

E. Fig. 1 shows that it is a complete bipartite graph because each of the vertices A, B, C from the first set V_1 are connected with each of the vertex D, E, F of the second set V_2 and that neither A, B, C nor D, E, F are adjacent to each other. In this paper, we have chosen a complete bipartite graph to represent genome sequences because nucleotide triplets can easily be expressed by nodes of the graph and the edges of the graph are used to calculate vector descriptor.

F.



G.

H.

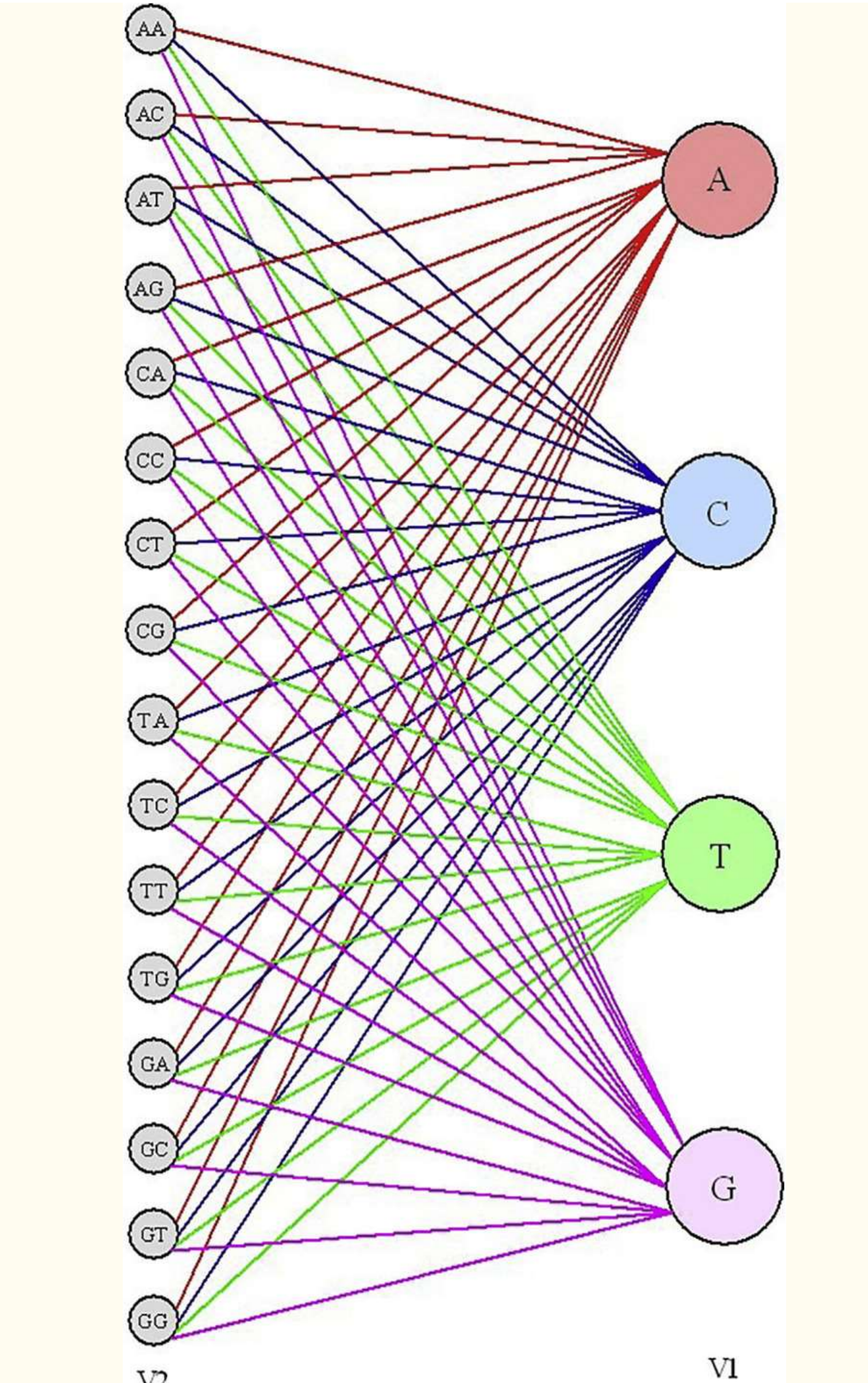
I. Fig. 1

J. Example of a complete bipartite graph.

K. 2.1. Construction of bipartite graph using nucleotide triplets

L. As shown in Fig. 2 , we consider two independent sets V1 and V2, where V1 consists of (A, C, T, G), the four nucleotide bases as vertices and V2 consists of (AA, AC, AT, AG, CA, CC, CT, CG, TA, TC, TT, TG, GA, GC, GT, GG), the sixteen di-nucleotides as vertices. All the vertices of set V_1 are connected with every vertex of set V_2. This way it becomes a complete bipartite graph.

M.



N.

O.

P. Open in a separate window

Q. Fig. 2

R. Representation of nucleotide triplets using a complete bipartite graph.

S. 2.2. Calculation of weighted vector

T. First of all, n-1 nucleotide triplets are considered from the DNA sequence of length n, in an overlapping manner. Then each nucleotide triplet is thought of as a combination of a nucleotide and a di-nucleotide. In this fashion, 64 nucleotide triplets are represented with the help of a bipartite graph. For the sake of calculation, we assume that the weight of each edge is one. With this assumption, we finally calculate the weighted vectors of 64 components from the given sequences.

U. 2.3. Computation of distance matrix

V. Let $S_1$and S_2 be two DNA sequences of two different species. Let x_i and y_i be the corresponding vector values of the sequence S_1 and S_2, where $i = 1, 2, 3, \ldots, 64$. Now the distance between two sequences S_1 and S_2 is calculated using the formula D(S1,S2)=∑641|xi−yi||64D(S1,S2)=∑641|xi−yi||64. Thus we get the similarity matrix for a set of DNA sequences S_1, S_2, S_3, S_4…. , S_m, where m is the number of sequences.

W. In our work, the distance is calculated between two 4 × 16 matrices. In mathematics, such distance is measured between two 64 dimensional vectors, because a 4 × 16 real matrix is homeomorphic to a 64-dimensional vector. So any metric on R^{64} is sufficient for the purpose. Naturally Manhattan metric is one such choice. Hence $d(X,\ Y) = \sum_{i=1}^{64}|x_i - y_i|$ may be a suitable one. But in the present case, x_i and y_i are not arbitrary real numbers. Rather they are the weighted components of the weighted matrix of a graph. Naturally, they are interrelated. In such a case the average of the Manhattan metric which is also a metric is found to be very much useful in matrix comparison. This has the standard name weighted distance(WD) and in this case, it is given by d(X,Y)=164∑64i=1|xi−yi|.

X. Machine Learning approach:

Summarization is the task of condensing a piece of text to a shorter version, reducing the size of the initial text while at the same time preserving key informational elements and the meaning of content. Since manual text summarization is a time expensive and generally laborious task, the automatization of the task is gaining increasing popularity and therefore constitutes a strong motivation for academic research.

There are important applications for text summarization in various NLP related tasks such as text classification, question answering, legal texts summarization, news summarization, and headline generation. Moreover, the generation of summaries can be integrated into these systems as an intermediate stage which helps to reduce the length of the document.

In the big data era, there has been an explosion in the amount of text data from a variety of sources. This volume of text is an inestimable source of information and knowledge which needs to be effectively summarized to be useful. This increasing availability of documents has demanded exhaustive research in the NLP area for automatic text summarization. Automatic text summarization is the task of producing a concise and fluent summary without any human help while preserving the meaning of the original text document.

It is very challenging, because when we as humans summarize a piece of text, we usually read it entirely to develop our understanding, and then write a summary highlighting its main points. Since computers lack human knowledge and language capability, it makes automatic text summarization a very difficult and non-trivial task.

Various models based on machine learning have been proposed for this task. Most of these approaches model this problem as a classification problem which outputs whether to include a sentence in the summary or not. Other approaches have used topic information, Latent Semantic Analysis (LSA), Sequence to Sequence models, Reinforcement Learning and Adversarial processes.

In general, there are two different approaches for automatic summarization: **extraction** and **abstraction**.

The extractive approach

Extractive summarization picks up sentences directly from the document based on a scoring function to form a coherent summary. This method work by identifying important sections of the text cropping out and stitch together portions of the content to produce a conden

Y. Text summarization with neural networks :

Z. Text summarization is a well-known task in natural language processing. In general, summarization refers to presenting data in a concise form, focusing on parts that convey facts and information, while preserving the meaning. To understand this better, we can think of summarization as a process to find a subset of data that contains all the information.

AA. With the exponential growth of the Internet, people often find themselves overwhelmed with data and information, which in most cases is textual. Due to this, automatic text summarization is becoming a necessity for various fields like search engines, business analysis, market review, academics, and more. Automatic text summarization means generating a summary of a document without any human intervention.

BB. This is broadly divided into two classes — extractive summarization and abstractive summarization. Extractive summarization picks up sentences directly from the original document depending on their importance, whereas abstractive summarization tries to produce a bottom-up summary using sentences or verbal annotations that might not be a part of the original document.

CC. Building an abstractive summary is a difficult task and involves complex language modelling. In this article we’re going to focus on extractive text summarization and how it can be done using a neural network.

DD. To understand the difference better, let’s consider the following paragraph-

EE. “Educators all around the world have fears about instituting large systemic changes, and sometimes those fears are well grounded. However, we cannot afford to ignore the possibilities that AI offers us for dramatically improving the student learning experience. People need to understand the role of AI in improving the face of education in the years to come. The resistance faced by this new technology has to decrease because AI can not only help the teachers be more productive, but also make them more responsive towards the needs of the students.”

FF. The extractive summary of this would be-

GG. *However, we cannot afford to ignore the possibilities that AI offers us for dramatically improving the student learning experience.The resistance faced by this new technology has to decrease because AI can not only help the teachers be more productive, but also make them more responsive towards the needs of the students.”*

HH. The abstractive summary of the same paragraph would be-

II. *“Educators will have to overcome their fears and reduce the resistance faced by AI to implement it in the education system because of the opportunities it offers.”*

JJ. As you can see, the abstractive summary is much shorter and to the point than the extractive summary.

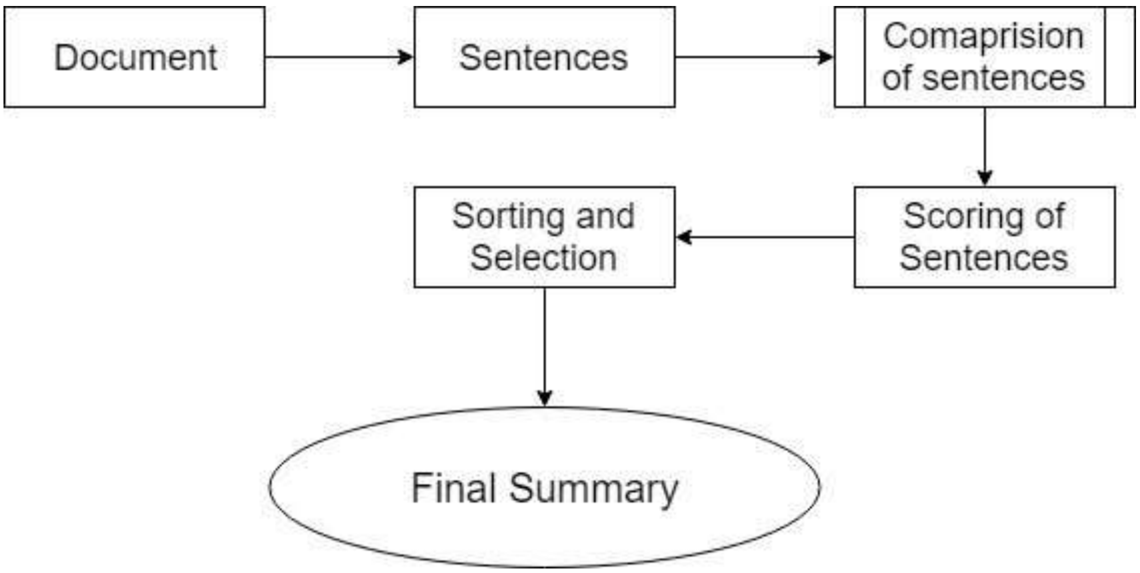
KK. Join more than 14,000 of your fellow machine learners and data scientists. Subscribe to the premier newsletter for all things deep learning.

LL. The Neural Network

MM. Until now, various models have been proposed for the task of extractive text summarization. Most of them have treated this as a classification problem that outputs whether a sentence should be included in the summary or not. They compare each sentence with every other one to select the most commonly-used words and give a score to each sentence on this basis.

NN. A threshold score is decided depending upon the length of the summary required, and every sentence having a higher score is then included in the summary. This is generally done using a Standard Naïve Bayes Classifier or Support Vector Machines.

OO. For genre-specific summarization (medical reports or news articles), engineering-based models or models that are trained using articles of the same genre have been more successful, but these techniques give poor results when used for general text summarization.



PP.

QQ. Flow Diagram depicting traditional models

RR. What if we could use a fully data-driven approach to train a feedforward neural network that gives reliable results irrespective of the genre of the document? A simple model consisting of one input layer, one hidden, and one output layer can be used for this task. This model would be able to generate a summary of arbitrarily-sized documents by breaking them into fixed-size parts and feeding them recursively to the network.

SS. If you see the paragraph we summarized earlier, you'll see that the summary contains the exact same sentences as in the original document were used:

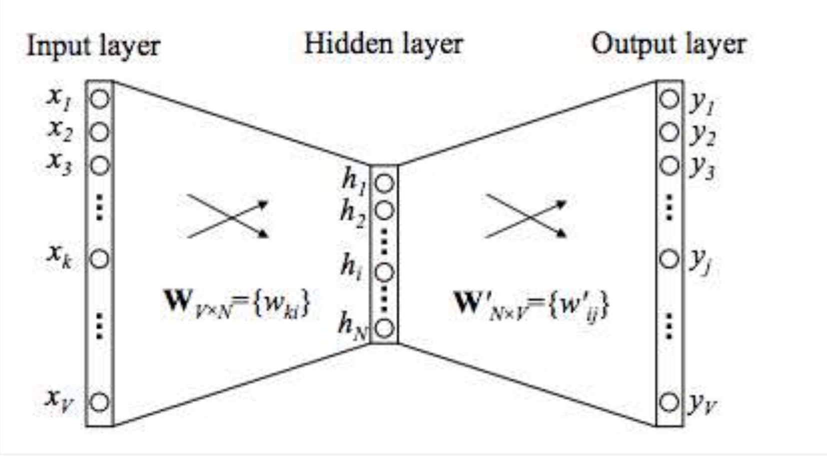
TT. *However, we cannot afford to ignore the possibilities that AI offers us for dramatically improving the student learning experience.The resistance faced by this new technology has to decrease because AI can not only help the teachers be more productive, but also make them more responsive towards the needs of the students.*

UU. These two sentences were selected by the model as most relevant to be included in the summary. The document is fed to the input layer, all the computation is done in the hidden layer, and an output is generated at the output layer as probability vectors, which determine whether a sentence is to be included into the summary or not.

vv. A fixed number of sentences are selected from every run depending on the size of the summary required. Since the input to the network must be numbers, we need a way to convert sentences to a numerical form.

ww. The best way of doing this is to convert the words to vector representations using the Word2Vec library. A tutorial for the same can be found here. Furthermore, these word vectors can be used to convert our sentences to vectors of fixed dimensions.

XX.



YY.

ZZ. A simple word2vec model

Automatic text summarization based on fuzzy logic :

In order to implement text summarization based on fuzzy logic, we used MATLAB since it is possible to simulate fuzzy logic in this software. To do so; first, we consider each characteristic of a text such as sentence length, similarity to little, similarity to key word and etc, which was mentioned in the previous part, as the input of fuzzy system. Then, we enter all the rules needed for summarization, in the knowledge base of this system (All those rules are formulated by several expends in this field

In Persian, the attributes used for choosing important sentences in the final summary are a little different from English. For example, in some cases, the last sentences in the paragraph or text have higher semantic value, while in Persian the first sentences have higher value and in many cases , these attributes are the same for both languages. In this paper, we changed the previous proposed fuzzy models [9] based on their application in Persian. Then, we implement and simulate this model again. 9. Simulation Results and Comparison In this experiment, we train all previously mentioned models on the twelve Persian features (using the same 100 Persian articles) and test these models by human judges to investigate the proposed system performance on a newswire data. Fig.3 shows the results of all models for the 100 Persian articles. Then we rank each document sentences based on this similarity value. A set of sentences is specified as a reference summary for each document based on the compression ratio. We chose 10 general Persian text to compare the result of Machine Learning method with fuzzy method. We gave these texts and the summaries produced by both Machine Learning and fuzzy methods to 5 judges who had an M.A. in teaching language . We asked the judge to read the main texts and to score the summaries produced by the two methods considering the degree to which they represent the main concepts. This means that if a user has to read one of these summaries instead of reading the main text, which summary conveys concept of the main text. The given score by the judges using the two methods are shown in table No.1. The results show that all the judge gave a better score to the summaries produced by fuzzy method. This indicates that fuzzy method worked better in parts of the sentence which contained uncertainty due to the use of fuzzy quantities. Therefore by using fuzzy approach in text summarization

9.Conclusion:

Automatic text summarization is an old challenge but the current research direction diverts towards emerging trends in biomedicine, product review, education domains, emails and blogs. This is due to the fact that there is information overload in these areas, especially on the World Wide Web.Automated summarization is an important area in NLP (Natural Language Processing) research. It consists of automatically creating a summary of one or more texts. The purpose of extractive document summarization is to automatically select a number of indicative sentences, passages, or paragraphs from the original document .Text summarization approaches based on Neural Network, Graph Theoretic, Fuzzy and Cluster have, to an extent, succeeded in making an effective summary of a document.Both extractive and abstractive methods have been researched. Most summarization techniques are based on extractive methods. Abstractive method is similar to summaries made by humans. Abstractive summarization as of now requires heavy machinery for language generation and is difficult to replicate into the domain specific areas.

View publication stats