# Triplet based Partitioning on Document Clustering

**Prof. Kalpana Deorukhkar, Ms. Anisa Tuscano, Ms. Malita Dodti**

**Abstract –**Now-a-days information retrieval plays a large part of our everyday lives – especially with the advent of the World Wide Web. During the last 10 years, the amount of information available in various forms especially the data in textual format represented in newspapers, journals, magazines, books etc. has grown exponentially. However, this development has introduced problems of its own; finding useful information is increasingly becoming a hit-or-miss experience that often ends in information overload. In this thesis, we propose document clustering as a possible solution for improving information retrieval on the Web. The primary objective of this project was to do topic modelling in which a group of words related to a particular topic are grouped so that it becomes easy for the user to improve the information retrieval. To achieve this end, two clustering algorithms that are Non-Negative Matrix Factorization (NMF) which uses a matrix representation and Inverse Log Frequency and Latent Dirichlet Allocation(LDA) which is based on a probabilistic approach to cluster the documents, were designed and implemented.

**Index Terms**— Clustering, LDA, NMF, n-gram modelling, TF-IDF Vectorizer, Tokenization, Word Sense Induction

## INTRODUCTION

Digital collections of data continue to grow exponentially as the information age continues to infiltrate every aspect of society. Today, most text data arrives in an unstructured form without any predefined organization or format, beyond natural language. The vocabulary, formatting, and quality of the text can vary significantly. Data clustering is an unsupervised data analysis and data mining technique, which offers refined and more abstract views to the inherent structure of a data set by partitioning it into a number of disjoint or overlapping groups. The main focus of the project is to help improving the quality and effectiveness of Word Sense Induction through clustering utilizing research within the area.

A system which can generate cluster of triplets from a document was being developed. Using the concept of similarity, sentences which are highly similar to each other are grouped into one cluster, thus generating a number of clusters. Here it implements two clustering algorithms such as Latent Dirichlet Allocation (LDA), Non-negative Matrix Factorization (NMF) also it compares the effectiveness and quality of result generated .

### 1.1. Drawbacks of the current approaches:

As mentioned in [1], imagine that you were given 100 newspaper articles and asked to sort them in a number of piles, reflecting their content. The number of piles and the central themes of the article piles are entirely up to you. You are also free to choose whether you want to read through every article or if you will only read the headings and skim through the contents. Such is the task of a document clustering system, with the only difference being that the task involves a lot more than 100 documents due to which it becomes difficult to find the appropriate data which the user wants at that point of time.

### 1.2. Proposed Solution to the above problem:

Topic modelling in which a bag of words is created by removing the stop words and stemming of the words so as to get the root word. This bag of words is then used to form cluster of words related to a particular topic and it is done by using two different algorithms such as :

**1.2.1. LDA** - This algorithm uses the probabilistic approach to find the target words. The probability of all the words is calculated, contained in the bag of words and then depending on the words having the highest probability it selects the target words and the triplets are formed by taking the adjacent words of the target word.

**1.2.2. NMF** - In this algorithm a document term matrix is created. Each document is represented by a vector in a m-dimensional co-ordinate space where m is number of unique terms across all documents. It then calculates TF-IDF scores and depending on that top 20 ranked terms are taken which gives us a very rough sense of the content of document collection. NMF produces to factor matrices as its output: W and H. The W factor contains the document membership weights relative to each of the k topics. Each row corresponds to a single document, and each column corresponds to a topic. We then provide the value of k and accordingly the number of clusters will be created. These clusters are represented using word cloud depending on the cluster number which the user wants to visualize at that time. Also a bar graph is represented which tells the weight of each term in the document.

The automatic discovery of document clusters/groups in a document collection, where the formed clusters have a high

degree of association (with regard to a given similarity measure) between members, whereas members from different clusters have a low degree of association.

In other words, the goal of a good document clustering scheme is to minimize intra-cluster distances between documents, while maximizing inter-cluster distances (using an appropriate distance measure between documents).

## 2. RELATED WORK

The use of word senses instead of word forms has been shown to improve performance in information retrieval, information extraction and machine translation. Word Sense Disambiguation [14] generally requires the use of large-scale manually annotated lexical resources. The triplets for the target word in each sentence are extracted, then use the related word to construct feature vectors for the sentence. Sense induction is typically treated as a clustering problem, by considering their co-occurring contexts, the instances of a target word are partitioned into classes.

Document clustering is used in many different contexts, such as exploration of structure in a document collection for knowledge discovery, dimensionality reduction for other tasks such as classification, clustering of search results for an alternative presentation to the ranked list and pseudo-relevance feedback in retrieval systems.(Christopher M. De Vries) The goal of clustering is to find structure in data to form groups. As a result there are many different models, learning algorithms, encoding of documents and similarity measures, which lead to different induction principles thus result in discovery of different clusters.

Cluster based method for Document Clustering [17] to identify the most important pieces of information from the document, omitting irrelevant information and minimizing details to generate a compact coherent summary document.

Partition clustering algorithms have been recognized to be more suitable as opposed to the hierarchical clustering schemes for processing large datasets.( Anna Huang) Accurate clustering requires a precise definition of the closeness between a pair of objects, in terms of either the pairwise similarity or distance. A variety of similarity or distance measures have been proposed and widely applied, such as cosine similarity and the Jaccard correlation coefficient. Measures such as Euclidean distance and relative entropy have been applied in clustering to calculate the pair-wise distances.

Dimensionality Reduction (DR) is a typical step in many text mining problems which involves transforming sparse data into a shorter and more compact one. DR can be done in two ways: feature reduction and feature selection.( A. Sudha Ramkumar, Dr. B. Poorna, 2016). Feature Selection dimensionality reduction selects a subset of the original representation attributes focusing on the word importance based on the evaluation function.

An unsupervised algorithm that can accurately disambiguate word senses in a large, completely untagged corpus [15]. The algorithm is based on two powerful constraints - that words tend to have one sense per discourse and one sense per collocation, Thus it avoids the need for costly hand-tagged training data.

Topic modelling is used to automatically discover the hidden thematic structure in a large corpus of text documents. A corpus of unstructured text documents is given as input. Two general approaches are popular: 1) Probabilistic approaches: View each document as a mixture of a small number of topics. Words and documents get probability scores for each topic. E.g. Latent Dirichlet Allocation (LDA). 2) Matrix factorization approaches: Apply methods from linear algebra to decompose a single matrix (e.g. document-term matrix) into a set of smaller matrices. For text data, we can interpret these as a topic model. e.g. Non-negative Matrix Factorization.

## 3. MODULE DESCRIPTION

### 3.1 Pre-processing:

In this module we take 4,551 documents from the news section and perform various steps to process this document as per the requirement of our algorithm. The pre-processing steps which is common to both algorithms is shown in Fig 1 and are discussed below.

3.1.1 Removal of Stop words - In computing, stop words are words which are filtered out before or after processing of natural language data. Though stop words usually refers to the most common words in a language, there is no single universal list of stop words used by all natural language processing tools. Any group of words can be chosen as the stop words for a given purpose. For some search engines, these are most common, short function words, such as the,

is, at, which, and on. In this algorithms non informative words from the documents are removed .The choice of stop words can have a considerable impact on the documents later on so uses a custom stop words list.

3.1.2 Stemming - Stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form generally a written word form. The stem need not be identical to the morphological root of the word; it Is usually sufficient that related words map to the same stem, even if the stem is not in itself a valid root. Eg: A stemming algorithm reduces the words “fishing”, ”fished”, and “fisher“ to the root word, ”fish”. In our algorithm we have used Snowball Stemmer.

3.1.3. Tokenization - Tokenization is the process of replacing sensitive data with unique identification symbols that retain all the essential information about the data without comprising its security. Tokenization seeks to minimize the amount the data which is needed for computation.
Two functions are defined - tokenize_and_stem: tokenizes (splits the synopsis into a list of its respective words (or tokens) and also stems each token. tokenize_only: tokenizes the text only.

3.1.4 Generation of N-Gram
In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sequence of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus. When the items are words, n-grams may also be called shingles. An n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram" (or, less commonly, a "diagram"); size 3 is a "trigram". Larger sizes are sometimes referred to by the value of n, e.g., "four-gram", "five-gram", and so on.

An n-gram model is a type of probabilistic language model for predicting the next item in such a sequence in the form of a (n − 1)–order Markov model. n-gram models are now widely used in probability, communication theory, computational linguistics (for instance, statistical natural language processing), Computational biology (for instance, biological sequence analysis), and data compression.


Fig 1. Flowchart for Document Clustering

Two benefits of n-gram models (and algorithms that use them) are simplicity and scalability – with larger n, a model can store more context with a well-understood space– time trade-off, enabling small experiments to scale up efficiently. N-grams can also be used for efficient approximate matching. By converting a sequence of items to a set of n-grams, it can be embedded in a vector space, thus allowing the sequence to be compared to other sequences in an efficient manner. For example, if it convert strings with only letters in the English alphabet into single character 3-grams, a -dimensional space (the first dimension measures the number of occurrences of "aaa", the second "aab", and so forth for all possible combinations of three letters)is obtained. Using this representation, information about the string is lost. For example, both the strings "abc" and "bca" give rise to exactly the same 2-gram "bc" (although {"ab", "bc"} is clearly not the same as {"bc", "ca"}). However, Empirically that if two strings of real text have a similar

vector representation (as measured by cosine distance) then they are likely to be similar.

- Once the pre-processing of data i.e stop words removal, lemmatisation/stemming, tokenisation is done.
- The frequency of words present in pre-processed data is calculated.
- High frequency words are taken as target words
- Here the n-gram count is specified where n=1,2,3
- Based on this number it uses algorithm to select n most frequent occurring features. This reduces computational cost to large scale and even saves memory
- For example: Target word to be disambiguated, e.g. space ,and the output is the number of word sets representing the various senses, e.g. (3-dimensional, expanse, locate) and (office, building, square).

**3.2 LDA**

In this algorithm a raw input of about 4,551 news articles are used. Steps are depicted in Fig. 2.

3.2.1 NLTK (Natural Language toolkit) package - NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

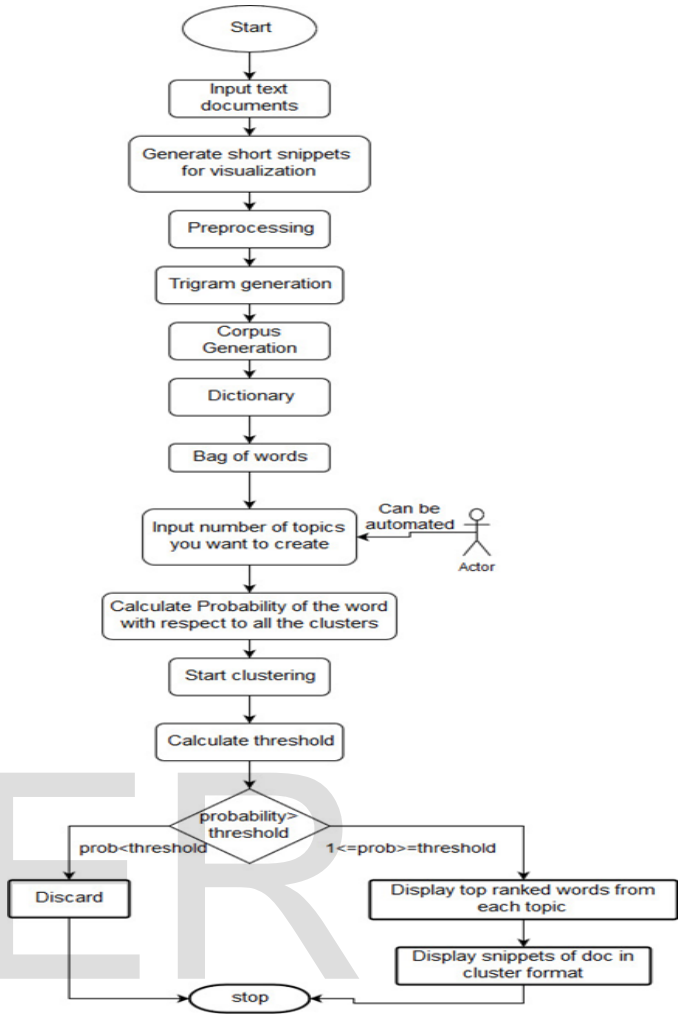


Fig 2. LDA flowchart

3.2.2 The words are then pre-processed to remove stop words, stemming to get the root word and then the remaining words are converted into tokens. the gensim package to calculate similarity between the words is used so that we can form triplets of the related words as shown in Fig 3.

Gensim package - Gensim is a Python library for topic modelling, document indexing and similarity retrieval with large corpora. Target audience is the natural language processing (NLP) and information retrieval (IR) community.

3.2.3. The triplets are the clustered to form a group which is called as bag of words. Then a set of topics are generated depending on the input given by the user. This topics include a group of words related to that topic.

3.2.4. Then the probability of each words in the topic is calculated and based on the probability calculation the

threshold is calculated by taking average of the probabilities. The words below threshold are discarded and then finally a cluster of related documents is formed.

```
for document1 in raw_documents1:
    trigrams = ngrams(document1.split(), 3)
    for grams in trigrams:
        #print(grams)
        trigram=' '.join(grams)
        #print(s)
        my_documents1.append(trigram)
```

```
texts = [[w] for w in my_documents1]

for i in range(20):
    print(texts[i])
```

```
['barclay defianc us']
['defianc us fine']
['us fine merit']
['fine merit barclay']
['merit barclay disgrac']
['barclay disgrac mani']
['disgrac mani way']
```

Fig 3. Formation of Triplets

3.2.5. The representation of this cluster is done as mentioned in [2], in which the left panel of our visualization presents a global view of the topic model. In this view, we plot the topics as circles in the two-dimensional plane whose centers are determined by computing the distance between topics. The right panel of our visualization depicts a horizontal bar chart whose bars represent the individual terms that are the most useful for interpreting the currently selected topic on the left. The left and right panels of our visualization are linked such that selecting a topic (on the left) reveals the most useful terms (on the right) for interpreting the selected topic as shown in Fig 4.

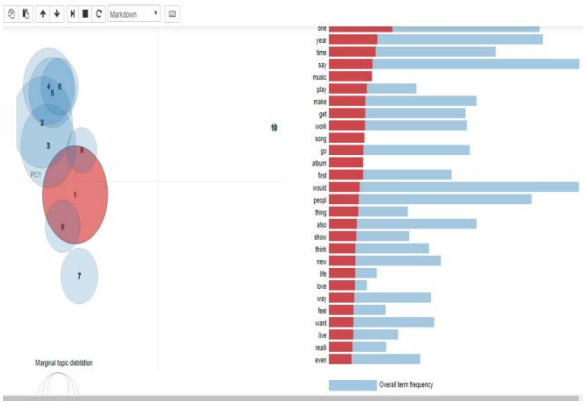

Fig 4. LDA Visualization

3.3 NMF Implementation

3.3.1 Topic modelling aims to automatically discover the hidden thematic structure in a large corpus of text documents. Steps are depicted in Fig 5. One approach for topic modelling is to apply *matrix factorization* methods, such as *Non-negative Matrix Factorization (NMF).*The dataset remains the same as that of the above mentioned algorithm.

3.3.2 After pre-processing the documents a bag-of-words model is formed in which, each document is represented by a vector in a m-dimensional coordinate space, where m is number of unique terms across all documents. This set of terms is called the corpus vocabulary.

3.3.3 As mentioned in [1] n-gram algorithm is used to extract trigram vectors from the bag of words. Since each document can be represented as a term vector, stack these vectors to create a full document-term matrix. The matrix from a list of document strings using Count Vectorizer from Scikit-learn [3] can be easily created.

3.3.4 To improve the usefulness of the document-term matrix by giving more weight to the more "important" terms the most common normalisation is term frequency–inverse document frequency (TF-IDF). The TF-IDF weighted document-term matrix is generated by using TF-IDF Vectorizer in place of Count Vectorizer as calculated in [3].

3.3.5 From the TF-IDF calculated the top ranked 20 terms are displayed, which gives a very rough sense of the content of the document collection. To find the similarity between the terms , the cosine distance as mentioned in [4] is calculated.

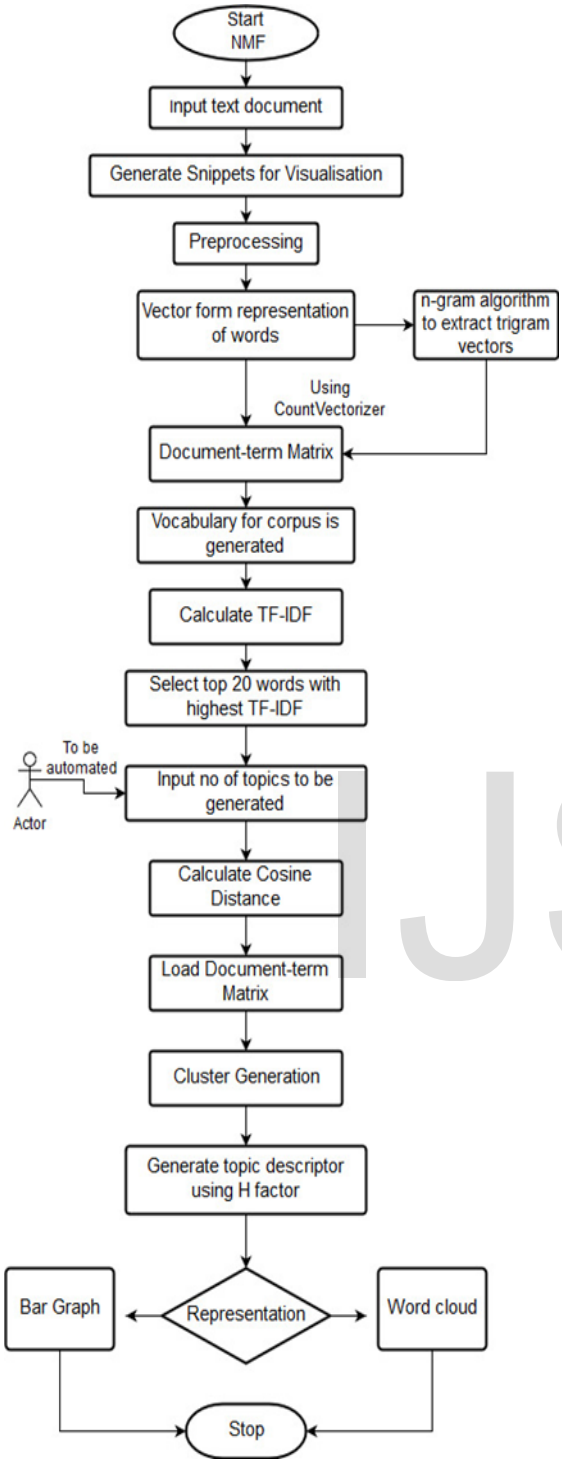


Fig 5. NMF Flowchart

3.3.6 Then the topics are generated depending on the inputs given by the user and the representation of each topic is done using word cloud and the frequency graph as given in Fig 6 and Fig 7 respectively.

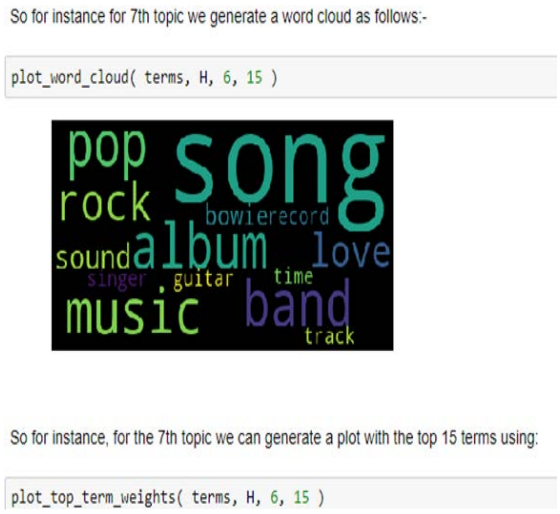


Fig 6: Word Cloud

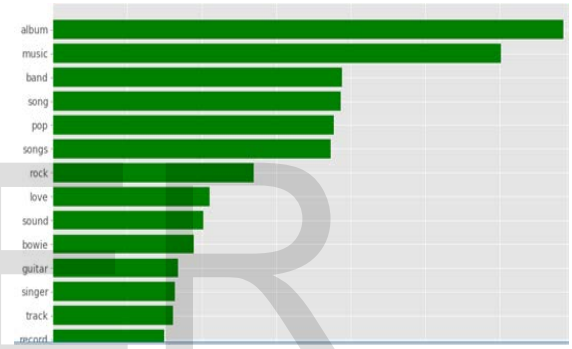


Fig 7. Word frequency in Word Cloud

## 4. Evaluation and Results

Evaluation measure used in NMF algorithm: Cosine Distance Calculation - When documents are represented as term vectors, the similarity of two documents corresponds to the correlation between the vectors. This is quantified as the cosine of the angle between vectors, that is, the so-called cosine similarity. Cosine similarity is one of the most popular similarity measure applied to text documents, such as in numerous information retrieval applications and clustering too. An important property of the cosine similarity is its independence of document length.

Evaluation measure used for LDA algorithm: Probability Calculation – When documents are represented as bag of words and the

number of topics is given by the user it then calculates the probability of the word with respect to all clusters. The calculation of probability plays a major role in clustering all the documents.

### 4.1 Accuracy Calculation

More commonly, it is a description of systematic errors, a measure of statistical bias; as these cause a difference between a result and a "true" value, ISO calls this trueness. Alternatively, ISO defines accuracy as describing a combination of both types of observational error above (random and systematic), so high accuracy requires both high precision and high trueness. In simplest terms, given a set of data points from repeated measurements of the same quantity, the set can be said to be precise if the values are close to each other, while the set can be said to be accurate if their average is close to the true value of the quantity being measured. The two concepts are independent of each other, so a particular set of data can be said to be either accurate, or precise, or both, or neither.

In the clustering algorithms different values of k (cluster number) are taken to calculate the accuracy of each cluster as shown in Fig 8. Accuracy is measured with respect to gold standard in which the true results are documented. So by comparing the accuracy of the algorithms with the gold standard an accuracy of 97% which is quite good is obtained.

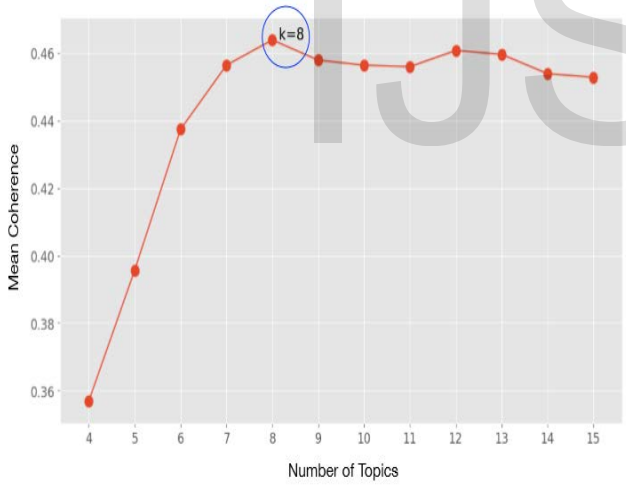


Fig 8.Accuracy Graph

### 4.2 Result Analysis

Two models of cross-situational learning of word meanings based on topic discovery algorithms, NMF and LDA are compared. Both models achieved high performance in every experimental cases when there is a set of sufficient learning samples. They proved to be robust to both linguistic and referential ambiguities and both models were able to support active learning which was shown to accelerate the learning speed by comparison with random sample selection. Each algorithm has its own better-suited scenario. NMF would be more adapted when dealing with only visual ambiguities and raw visual data ("keywords only" scenario), resulting in precise mono-modal concepts, once a correct number of components is provided. LDA shows better adaptability and robustness with clustered visual data when linguistic ambiguity and noise are involved ("full sentence" scenario) due to its statistics-based nature. Contrary to this embedded mechanism of keywords selection in LDA, NMF has to be associated with a language filtering mechanism but is not able to reach similar performances in the "full sentence" scenario. Yet from our implementation, the resulting strategy is different: random sample choices in the "triple" scenario led to a mean repetition of 2.42 words in successive steps, while the active choice led to a mean repetition of 1.89 words. Two basic reasons could be used to explain such a difference in applying the repetition strategy. On one hand, in [15], each trial consists of four mutually different objects thus no "within-trial repetition of objects" is allowed, however in our "triple" scenario experiment, the same features (shape or colour) from different objects could appear in a triple and this gives rise to a "within-triple feature repetition" which can simply reduce the complexity of each triple. In fact, the number of repeated features inside a triple is 0.86 with the random strategy and 2.06 with the active choice. On the other hand, unlike computational models, humans are less efficient at keeping a long-term memory of the past co-occurring records and hence the successive repetition facilitates learning.

## 5. Conclusion

The goal of text document clustering is to minimize the intra cluster distance between documents while maximizing the inter cluster distance using an appropriate distance measure between documents. With the full data representation, By applying the Feature Selection Dimension Reduction method the LDA and NMF algorithm can effectively clusters the document collection thus improves the clustering quality significantly.

The algorithms works by harnessing several powerful, empirically-observed properties of language, namely the strong tendency for words to exhibit only one sense per collocation and per dis- course. It thus uses more discriminating information than available to algorithms treating documents as bags of words, ignoring relative position and sequence. Indeed, one of the strengths of this

work is that it is sensitive to a wider range of language detail than typically captured in statistical sense-disambiguation algorithms.

According to the experiments results, we can conclude that there is a better performance in terms of response time, memory utilization and processing distribution comparing with previous method used.

**6 Future Enhancements**

In the algorithms implemented the number of clusters are taken from the user, but in future this number of clusters can be automated as per the number of documents and the topics containing each documents. For the implementation of the algorithms the articles are used in their text format which belongs to the news section. In future this dataset can be changed. Since the clusters can be automated the algorithms will generate the best possible number of clusters and it will be very efficient for clustering the documents related to a specific topic. LDA takes a longer time for its implementation since the dataset is not properly pre-processed. In future the computation time can be reduced by adding more pre-processing functions so that the data is very well pre-processed to avoid any ambiguity thereby reducing computation time. The words in the cluster can be overlapped sometimes. In future we can remove this overlapping by proper disambiguation methods.

## 6. References

[1] Information Retrival In Document Spaces using Clustering by Kenneth Lolk Vester and Moses Claus Martiny.

[2] LDAvis: A method for visualizing and interpreting topics Carson Sievert Iowa, State University USA Kenneth E. Shirley, AT&T Labs Research NewYork,NY10007,USA

[3] Topic Modelling with SciKit-Learn Derek Greene, University College of Dublin

[4] Similarity Measures for Text Document Clustering Anna Huang Department of Computer Science The University of Waikato, Hamilton, New Zealand.

[5] Search Behavior Based Latent Semantic User Segmentation for Advertising Targeting Xueqing Gong, Xinyu Guo, Rong Zhang, Xiaofeng He∗ and Aoying Zhou Software Engineering Institute East China Normal University, Shanghai, China

[6] Topic Modelling with SciKit-Learn Derek Greene,University College of Dublin

[7] Discovering Corpus-Specific Word Senses Beate Dorow, Institute for Maschinelle Sprachverarbeitung Universita Stuttgart, Germany Dominic Widdows, Center for the Study of Language and Information Stanford University, California.

[8] Document Clustering Using Cluster Based Method Archana A.B1, Sunitha .C2, Anoop S Babu3, Sandra Sarasan Calicut University , Vidya Academy of Science and Technology ,Amrutha Vishwa Vidyapeetham & Amrutha School of Engineering.

[9] An experimental comparison between NMF and LDA for active cross-situational object word learning. Yuxin Chen, Jean-Baptiste Bordes, David Filliat

[10] Seman&c Analysis in Language Technology: Word Sense Disambiguation Marina Santini Department of Linguistcs and Philology Uppsala University, Uppsala, Sweden

[11] A Fully Unsupervised Word Sense Disambiguation Method Using Dependency Knowledge: Ping Chen, Wei Ding, Chris Bowes, David Brown ; Dept. of Computer and Math. Sciences University of Houston-Downtown

[12] A Survey paper on different techniques of document clustering: 1Mamta Mahilane, 2Mr. K. L. Sinha ;1M.Tech Scholar, 2Sr. Assistant Professor Department of Comp. Sci. & Engg. Chhatrapati Shivaji Institute of Technology, Durg Chhattisgarh, India

[13] Word Sense Induction: Triplet-Based Clustering and Automatic Evaluation Stefan Bordag, Natural Language Processing Department University of Leipzig, Germany

[14] Triplet-Based Chinese Word Sense Induction
Zhao Liu, Xipeng Qiu, Xuanjing Huang, 2010.

[15] Unsupervised word sense disambiguation rivaling supervised methods, David Yarowsky

[16] Document clustering using Cluster Based Methods, Archana A.B, Sunitha .C, Anoop S Babu, Sandra Sarasan, IJETAE , Volume 3, Issue 7, July 2013