Latent Dirichlet Allocation for Test of English as a Foreign Language Test

Kajarekar Sunit Pravin Aruna
Computer Science Dept., Illinois Institute of Technology
Chicago, Illinois, United States
skajarekar@hawk.iit.edu

Sukhada Pande
Computer Science Dept., Illinois Institute of Technology
Chicago, Illinois, United States
spande2@hawk.iit.edu

Nikita Khedekar
Computer Science Dept., Illinois Institute of Technology
Chicago, Illinois, United States
nkhedekar@hawk.iit.edu

Zhenghao Zhao
Computer Science Dept., Illinois Institute of Technology
Chicago, Illinois, United States
zzhao482@hawk.iit.edu

Abstract*—Topic modelling is great unsupervised process of data mining for opinion mining, text mining described as method which discovers the hidden topics from the collection of documents that best represent the information in large collection. This paper reviews our attempts to automatically identify topics present in a text corpus and to drive hidden patterns revealed by a text corpus using topic modelling(used to evaluate TOEFL test questions), and explains it as a better decision making process. Here Latent Dirichlet Allocation (LDA) is applied as a topic model to set of documents and split them into topics. The reliability of topic models is examined by using the different similarity measures. We have tested LDA against Test of English as a Foreign Language (TOEFL) synonym questions. With the help of LDA we achieved dimensionality reduction; topic best fitted is with lowest dimensions i.e. 50 dimensions. We have also explored word embedding another type of topic modelling technique which is one of the most popular representation of document vocabulary that capture their meanings, sematic relationships and different types of context they are used in. Finally, compared results obtained from LDA model and word embedding. We got better accuracy for word embedding than LDA with less vocabulary but large number of questions were discarded.

Keywords—*Latent Dirichlet Allocation (LDA), word embeddings, Topic modelling*

I. INTRODUCTION

This course project is a part of the 4Humanities "WhatEvery1Says" project (WE1S) [8] Based at the University of California, Santa Barbara (UCSB), with core collaborators at California State University, Northridge (CSUN) and University of Miami (UM). WE1S investigates how the news media and other public sources depict the humanities. It uses LDA topic modelling to study public communication on the humanities in journalistic media and other sources at large data scales. It imitates the main themes, frames of discussion, and narratives of such discourse, the relations between these and also unexpected themes or relations. For example here, to find how do journalists, politicians, business people, scientists, parents, students, university administrators, professors, writers, artists, and others typically talk about the "humanities". Also it includes study of how public discourse on the humanities compares across states, nations, and regions of the world. Earlier it was engaged in a small-scale pilot project for this purpose. Then WE1S was funded by Mellon Foundation so that, scope and diversity of data can be expanded remarkably, to increase range and trustworthiness of analytical methods used.

The main computational process that WE1S applies to analyze its collected materials is topic modeling. It is a leading method of machine-learning analysis, topic modeling discovers through statistical means the existence, relative weight, and distribution of "topics" across documents. Topic modeling can be particularly important for discovering areas of public discourse related to the humanities that are not colored by preconceived theses or expectations. Moreover, WE1S will explore "word embedding" (word2vec) and text-classification. In our project we are using LDA to find topic with good dimension which is best fitted. We then measured performance of model using different similarity measures and then compared it with word embeddings to evaluate our model. [5] [1]

II. PREVIOUS WORK

There have been previously a lot of work done on determining or finding answers to synonym questions using different dimensionality reduction techniques like LSA and LDA. Here are the few of them.

In paper by Derrick Higgins , he uses LSA to determine the synonym for toefl questions, and also introduces a new similarity measure i.e. word-similarity scores(LC-IR) which outperforms many methods, he uses the approach of distribution of synonyms. With his approach he got accuracy of 81.25 % correctness.

Similarly, a paper by Thomas K. Lindauer on Latent semantic analysis theory also uses LSA as dimensionality reduction technique for determining answers to linguistic

questions. The model got 51.5 correct, or 64.4% (52.5% corrected for guessing).[6][7]

There has being previous work of using LDA as an approach to find answers to Toefl synonym questions that in paper by Vasiljevic et al. where they have used LDA to answer the questions and have used cosine similarity measures to find similarity between the words using LDA topic modelling with various topic modelling of topics from 66,357, 2766, 4576 etc[3][4]

III. DATA:

A. *Origin/Purpose*

The 4humanities WhatEvery1Says project (WE1S) uses digital humanities methods to study public discourse about the humanities at large data scales. The project scope concentrates on but is not limited to, news articles that have been produced from the New York Times in digital format between 1980 and 2017. We have mallet LDA outputs for a different number of topics with vocabulary size of 150,000 words. In addition to that, for word embedding (300 dimensions) used vectors_300.tsv (10000 X 300) and metadata_300.tsv (300 X 1) files which provides document vocabulary representation with vocabulary of almost 10000 words. The vocabulary size of LDA is greater than word embedding.

B. *Source of availability, formats*

We have used Dataset-B, collected from news articles that contain the keyword "humanities" and were digitally created by the New York Times between 1980 and 2017. It is collected and digitized from the New York Times and hence has less typing errors. The data contents and format,

Contents	**Format**
It contains information about all data files in short. It also contains copyright and licensing information.	0_Readme ad-hoc text format
Mallet LDA model files for a different number of topics (50, 100, 200, 800, and 1600)	Dataset-B-no_of_topictopics.zip Ex: Dataset-B-50topics.zip
Word Embedding matrix and metadata.	Dataset-B_WordEmbedding 300.zip vectors_300.tsv(10000X300) metadata_300.tsv (300X1)

Table1: Data Format

We parsed mallet output (topic-state.gz) to predict synonym multiple choice questions of TOEFL test. Word embedding (300 dimensions) obtained using skip-gram method with window size of 5. In vectors_300.tsv each row represent word vector of 300 dimensions. Metadata_300.tsv contains metadata for vectors_300.tsv. Each line corresponds to a row in the matrix. Note that the matrix and metadata are sorted with the most frequent words on top.

Topic models[1] learns from training data so for small set with very less diversity of topics it is hard to test this classifier. Here we can see domain or topic models trained on New York Times news data with keyword "humanities".[8] The testing done with LDA on TOEFL test gives fairly good results hence we can say that the test topics are diverse (matching training data). Word embeddings are trained on same data and used to derive similarities and relations between words. As data used is same the context of word embeddings matches with that of data.

C. *TOEFL Data*

The synonymy test represents a list of words and for each of them, there are 4 candidate words. The task is to determine which of these candidate words which is synonym to the word in question is.There are two types of questions, First is Simple Synonymy questions And other is TOEFL synonym question format is synonym questions based on context, A word might have more than one meaning, but the meaning of the word in given context is relevant. We have used questions from the TOEFL sample question data which is available at

https://aclweb.org/aclwiki/TOEFL_Synonym_Questions_(State_of_the_art). There are total 80 questions on which we are evaluating our topic models.

Some of the tools, Packages and Libraries used are Python, Jupyter, Mallet (Mallet is an open source package that can be used for topic modelling, document clustering and NLP), Tomas Mikolov's Word2vec, numphy (performing calculations), Scikit, pandas (Processing and creating dataframes used for data manipulations and analysis),SciPy, matplotlib (visualization).

IV. GOALS AND OBJECTIVES

The primary goal of this project is to check how effectively LDA and word embedding topic models built on New York Times news with key word "Humanities" data predicts TOEFL test questions.

In future, we can use this data in several other ways as it contains broad spectrum of phrases such as "Humanities", "liberal arts" and "arts" as the data we are getting from New York Times news is very interesting and diverse.[8] Topic models are very beneficial for the purpose of document clustering, organizing large blocks of textual data, information retrieval from unstructured text and feature selection. For Example – New York Times are using topic models to boost their user – article recommendation engines. Various professionals are using topic models for recruitment industries where they aim to extract latent features of job descriptions and map them to right candidates. They are being used to organize large datasets of emails, customer reviews, and user social media profiles. It is so interesting that we can use topic models for creating several context based word predicting applications (eg. Grammarly) which gives suggestions when you place wrong word at any place while writing or you are doing any grammatical mistake moreover sends you reports on weekly basis of the edits you carried out with small analysis and mistakes you are done by

mail. Also these topic models can be used for predicting effect of any particular events (elections/ recession/ any new scheme /new product in market).

V. APPROACHES

A. *LDA*

Topic modelling is a type of statistical modeling for discovering the abstract "topics" that occur in a collection of documents. LDA is an example of topic model and is used to discover hidden topics that occur in a collection of documents. These topics will only emerge during the topic modelling process therefore called latent. Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.[2]

LDA represents topics by word probabilities. The words with highest probabilities in each topic usually give a good idea of what the topic is can word probabilities from LDA.

What LDA says is that each word in each document comes from a topic and the topic is selected from a per-document distribution over topics. So we have two matrices:

1. Θtd = P(t|d) which is the probability distribution of topics in documents

2. Φwt = P(w|t) which is the probability distribution of words in topics

And, we can say that the probability of a word given document i.e. P(w|d) is equal to:

$$\sum_{t \epsilon T} p(w|t,d)\, p(t|d)$$

B. *Word Embedding*

There are two novel model architectures for computing continuous vector representations of words from very large data sets, CBOW (Continuous Bag of Word) or skip gram (Mikolov et al.) approach. Word embedding is a necessary step in performing efficient natural language processing in your machine learning models. Word embedding approach use the intuition that, "Similar word will also be surrounded by a similar context".

Word embeddings can be trained and used to find similarities and relations between words. If we have a document or documents that we are using to try to train some sort of natural language machine learning system, we need to create a vocabulary of the most common words in that document. Word embedding try to "compress" large one-hot word vectors into much smaller vectors (here 300 dimensions) which preserve some of the meaning and context of the word. For words which have similar contexts share meaning under Word2Vec, and their reduced vector representations will be similar. Here is the example of it from the given model (only upto 2 dimensions).

given	-0.15220729	-0.40800023
data	0.00502476	-0.6731983
orginated	0.407729	1.0964631
from	-0.09379256	-0.014016383
new	-0.10420346	0.079566434
york	-0.0957012	0.29372135
times	-0.050794285	0.26229668

We can see from above "new" "york" and "times" words has very different meaning but have similar vectors as the context in which they are used is similar.

Skip-gram use a word to predict the surrounding ones in window. In this approach we are calculating matrix which is a vector representations of each word in our vocabulary. More formally, given a sequence of training words w1, w2, w3… wT the objective of the Skip-gram model is to maximize the average log probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} \log p(w_{t+j}|w_t)$$

Where c is the size of the training context (which can be a function of the center word wt). Larger c results in more training examples and thus can lead to a higher accuracy, at the expense of the training time.

We need a way of ensuring that, words which are similar end up having similar embedding vectors. Therefore, we need a similar vectors outputting a high score and un-similar vectors outputting a low score. So for testing TOEFL questions with word embedding model by using similar similarity measures is great idea as used in approach 1.

C. *Similarity measures used*

The similarity measure is the measure of how much alike two data objects are. Similarity measure in a data mining context is a distance with dimensions representing features of the objects. If this distance is small, it will be the high degree of similarity where large distance will be the low degree of similarity. The similarity is subjective and is highly dependent on the domain and application. For example, two fruits are similar because of color or size or taste.[5]

C1. *Cosine Distance*

Then the cosine distance between two word vectors is the angle that the vectors to that word make. Larger (+ve) cosine means a smaller angle hence smaller distance and two words are similar. As we need to find the closest word we are using cosine distance as a similarity measure. Below is the formula given for cosine similarity and distance, where Ai and Bi are components of vector A and B.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Cosine Distance =1- cosine similarity

C2. Jensen - Shannon Divergence

In probability theory and statistics, the Jensen–Shannon divergence is a method of measuring the similarity between two probability distributions. Assume set $M_{+}^{1}(P)$of probability distributions where A is a set provided with some σ-algebra of measurable subsets,

$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2} D(P \parallel M) + \frac{1}{2} D(Q \parallel M)$$

$$\text{where } M = \frac{1}{2}(P + Q)$$

C3. Euclidean Distance

Euclidean Distance between two words is the shortest distance among them. If it is 0, it means that words are identical. Euclidean distance is the most common use of distance. In most cases when people said about distance, they will refer to Euclidean distance. Euclidean distance is also known as simply distance. When data is dense or continuous, this is the best proximity measure. The Euclidean distance between two points is the length of the path connecting them. The Pythagorean Theorem gives this distance between two points. Euclidean distance can be calculated as,

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

C4. Cosine Similarity

Cosine similarity metric finds the normalized dot product of the two word vectors. By determining the cosine similarity, we would effectively try to find the cosine of the angle between the two objects. The cosine of 0° is 1, and it is less than 1 for any other angle. It is thus a judgement of orientation and not magnitude: two vectors with the same orientation have a cosine similarity of 1, two vectors at 90° have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. Cosine similarity is particularly used in positive space, where the outcome is neatly bounded in [0,1]. One of the reasons for the popularity of cosine similarity is that it is very efficient to evaluate, especially for sparse vectors.

$$sim(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

VI. EVALUATION

A. Experiments Set-Up

A1. Data Preprocessing

Understanding preprocessing steps performed on data is a crucial step. The setting used in preprocessing tools while training should be in synchronization to setting used for testing to achieve good results. The steps performed on data,

i. Digital news articles from NY Times is subjected to preprocessing techniques of scrubbing, punctuation cleaning and stop words removal i.e. removing words like ‘a’, ‘an’ and ‘the’.

ii. This dataset is then converted to JSON files; one file per news article.

iii. BoW is extracted from these JSON files.

A2. LDA Model Costruction

The algorithm for topic modelling has been implemented by a software toolkit called ‘Mallet”. Mallet is an open source package that can be used for topic modelling, document clustering and NLP [3] [4].

The preprocessed data converted into analytical data for machine learning processes such as topic modeling. The text was run through Topic Modelling Tool- a Java implementation of MALLET tool for LDA topic modelling. The output of LDA MALLET is LDA model in different dimensions such as 50,100,200,400,800 and 1600. [4]

A3. Word embedding generation

Word embedding calculated for given data with “skip-grams” method with window size of 5, for 300 dimensions (embedding size) and top 10000 words. Vactor_300.tsv is a vector representations of each word in our vocabulary. We are using Vactor_300.tsv and metadata_300.tsv (one-hot vector representation) files as input to get word vector. In this approach we have created a matrix with dimension 10000 X 300 which is a vector representations of each word in our vocabulary.

A4. Parameter tunning

Normalization is used to make something normal or standard. We have normalized word-topic matrix using L2-norms to get “unit vector” for question and answer word for term topic matrix and improve accuracies obtained by LDA model. L2-norm is also known as least squares. It is basically minimizing the sum of the square of the differences between the target value and the estimated values. L2-norm is used as it is stable solution.

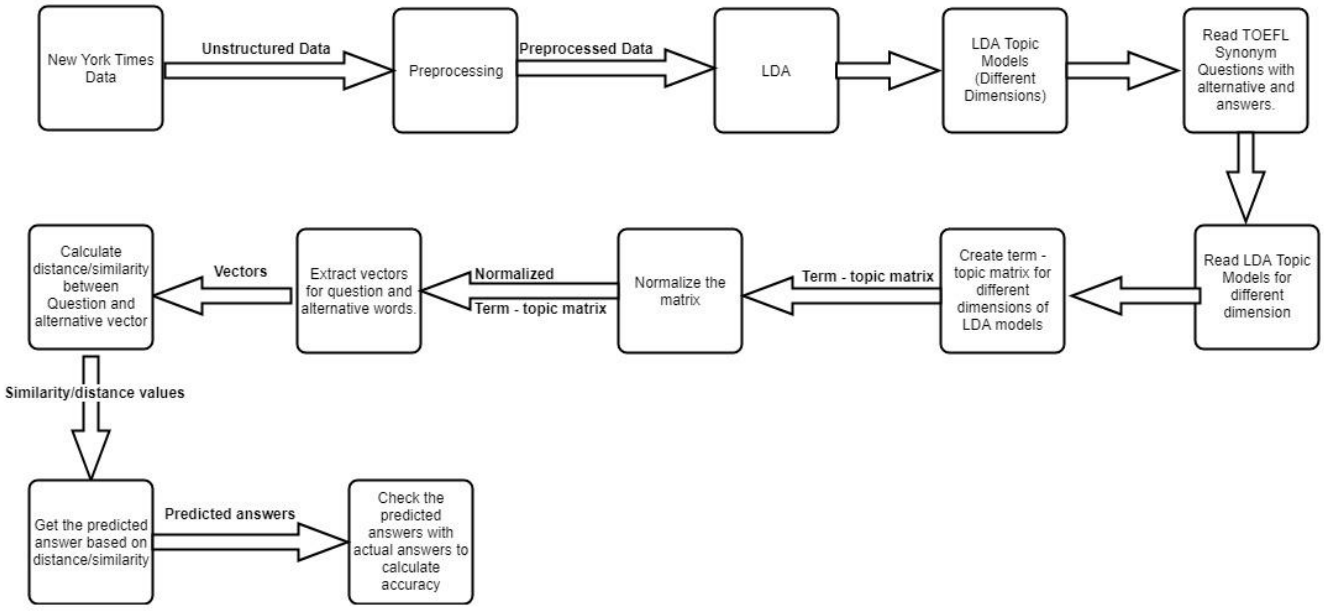

Figure 1 : Flowcharts for Experimental Steps

B. Experiments

The flow experiments performed can be described by Figure 1. The detailed description of steps are explained in this section.First we read TOEFL Synonymy questions data and answers data. For TOEFL question and answer we created lists in which, we store question word as a key and created list alternative_List which stores all alternative answer words for a specific question. As explained in approach1 we parsed the topic-state.gz file from the output of LDA application in mallet (data of trained LDA model) to predict TOEFL questions answers. The topic-state.gz file contains column-word index, word, Topic to which word belongs, using these we have formed Word-Topic matrix (words are rows and topics are columns) so the largest word index will be the dimension of the matrix formed and columns represent reduced topic dimensions. Word-Topic Matrix (pxq) where p is the word index (number of times word (p) appears in topic (q)) and q is the topic number. In addition to get better accuracies we have normalized our Word-Topic Matrix with reference to A4 parameter tuning. Then we found vectors of the questions and each alternative words. After this, we are using four different similarity measures to calculate similarity/distance between question vector and all four alternative vectors. The similarity to see how well the model predicts the TOEFL question's answer. We have performed these experiments on simple synonym questions to find the closest row (word vector) to question word from answer words. The correct answer choice is one having small distance or large values of similarity. Also calculating accuracy in order to test performance of model and to check which one best fitted amongst all. In addition to the steps above, we have experimented all the above experiments by discarding missing values. In this approach, we have checked missing answer choices in addition to missing question words in the word-index dictionary. Then we are discarding questions which either contains missing question word or missing answer choice from the total number of questions. At last, we calculated accuracies. This approach helped us to find the better values of accuracies of the model. We performed these steps for all dimensions.

In 2nd approach that is word embeddings, we have created a matrix with dimension 10000X300 which is a vector representation of each word in our vocabulary. Word embedding will provide higher similarity if the two words have a similar context. Word Vector created with the help of numpy.matrix. As explained in approach 2 we have used word embeddings output files as input to get a matrix which is a vector representation of each word in our vocabulary. So, each word (row) is represented by a vector of size 300. And then testing TOEFL questions with word embeddings model by using similar similarity measures used in approach 1. Also calculating accuracy and compared results of approach 2 with approach 1.

C. Results

C1. LDA MODEL (16 Questions Discarded)

Dimension	Cosine distance	Cosine similarity	Jensen Shannon Divergences	Euclidean distance
50	48.4375	48.4375	46.875	50
100	29.6875	29.6875	28.125	29.6875
200	35.9375	35.9375	32.8125	35.9375
400	42.1875	42.1875	42.1875	40.625
800	40.625	40.625	34.375	39.0625
1600	43.75	43.75	35.9375	39.0625

Table2: Accuracies of LDA model performed

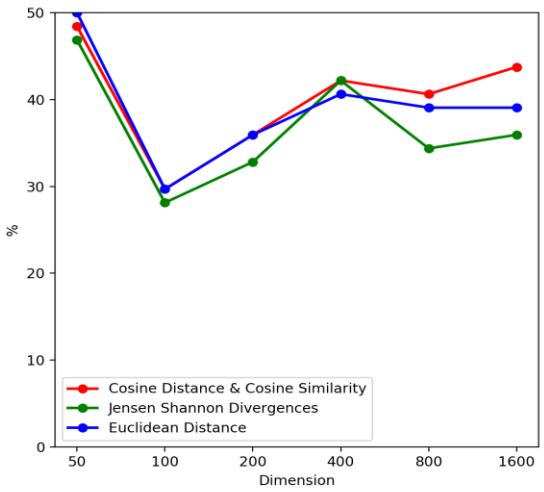

Figure2 : comparison between accuracies obtained from different similarity

C2. Word embedding accuracies (63 Questions Discarded)

Dimension	Cosine distance	Cosine similarity	Jensen Shannon Divergences	Euclidean distance
300	47.0588	47.0588	35.2941	35.2941

Table3: Accuracies of word embedding model performed

D. Analysis and colclusion

We need to find a model which fits to predict the TOEFL test questions. We should check this so that we will not get under-fitted or over-fitted model. In above tables we have same values for accuracies calculated by cosine similarity and cosine distances as cosine distance = 1-cosine similarity (inverse of each other) and both measures are predicting same number of questions. From table 2 and table 3,

D1. LDA

We can see LDA performed well for the 50-dimensional model with Euclidean distance similarity measures and got an accuracy of 50%. We got this result after discarding 16 questions in which question and alternative option words were missing. These questions are discarded so as to avoid false positive results. So we can conclude that LDA is used for dimensionality reduction. For every similarity measure, maximum accuracy is observed for the lowest dimension that is 50.

D2. Word Embedding

For word embedding, we got better accuracies than the LDA model. We got almost 47.0588% accuracy by using Cosine Distance and Cosine Similarity. We got this result after discarding 63 questions having question and alternative option words were missing. These are not good results as we have discarded a large number of questions and getting very fewer question prediction correct as compared to LDA. But this can be justified as vocabulary size of word embedding that is 10000 is very less than vocabulary size of LDA that is 150000.

E. Error analysis

Why we are not getting better results for word embedding? Also why we are getting 50% accuracy and not more than that for LDA? These questions can be answered by doing error analysis. For analyzing error we are appending our results obtained from our models along with question word, answer choices and actual answer into csv file for the respective model (LDA / Word Embedding) for all dimensions. We analyzed the performance of LDA model for 4 different parts of speech (noun, verb adjective and adverb). If all 6 dimensions predicts resulted well we have shown in bar graph "good performance". If none or one is predicted then it is shown in bar graph as "bad performance". And "mixed performance" means half predicted as right. After analyzing we learn that, we are getting better results for noun based words for 50 topic model. Bar graph below is representing performance of LDA model for different parts of speech.

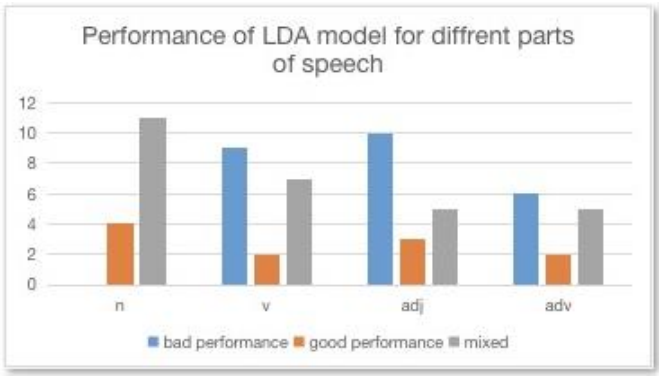

Figure3: performance of LDA for POS

	BAD performance	GOOD performance	Mixed	total
Noun	0	4	11	15
Verb	9	2	7	18
Adjective	10	3	5	19
Adverb	6	2	5	13

Table4: performance for different parts of speech

We also examined the similarity/distance values for the alternative words with respect to the question to check our algorithm. For e.g. prediction for word 'good' as 'nice' is acceptable whereas a prediction of 'bad' or 'horrible' is completely wrong. To analyze this situation, we considered a question word 'consumed' from our TOEFL question dataset and calculated the cosine distance between all the alternatives shown in the table below.

	Number of correct predictions	Number of questions after discarding	Accuracy	Number of discarded questions
noun	10	15	66.7	2
verb	7	18	38.9	0
Adjectives	8	18	44.4	8
adverbs	6	13	46.2	6

Table5: performance and accuracies for different parts of speech

The actual answer for the question word consumed is 'eaten'. But it can be seen from the table that eaten had the highest distance while supplied which is the predicted answer had the lowest. For the purpose of analysis, if we consider the distance of the actual answer to be 0.5 higher than the predicted answer; in this case it is (0.9447 – 0.2152 = 0.7295) as 'huge distance error'; we have around 12 questions out of the 31 wrong predictions i.e. 39% predicted by 50 dimensional topic model with 'huge distance error'.

Supplied	0.2152
eaten	0.9447
caught	0.3482
bred	0.2974

Table6: Cosine distance with word "consumed"

VII. Conclusion

The experiments performed proved that the results of topic modelling using LDA did perform relatively well in predicting the TOEFL test answers. To avoid false positives in our results we discarded those questions. Some of the questions that were discarded in our results are non-stemmed words. The accuracy of LDA depends on the corpus on which LDA models were trained on. On testing the same algorithm for a different dataset (University wire dataset with vocab size: 15348 and 250 dimensions and reddit dataset with vocab size: 24588 and 100 dimensions) an accuracy of 53%-54% was achieved which is higher than dataset used by us. For word embeddings method, as discussed in analysis a large number of questions were discarded due to the vocabulary size being much smaller than LDA models (10,000 compared to 150,000). For future scope, we can use Google word vectors having vocabular size of around 3 billion words to test our word embeddings approach. Also, we have reserved testing this LDA models against contextual questions or multiple word questions (Graduate Record Examination (GRE) questions) as our future work. Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

References

[1] Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y. and Zhao, L. (2018). Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. Multimedia Tools and Applications. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] Latent Dirichlet Allocation David M. Blei, Andrew Y. Ng, Michael I. Jordan; 3(Jan):993-1022, 2003.

[3] A. McCallum. (2002)., MALLET: A Machine Learning for Language Toolkit [online]. Available http://mallet.cs.umass.edu

[4] D. Mimno, "Machine Learning with MALLET," Department of Information Science, Cornell University.

[5] Vasiljević, J., Ivanović, M., Lampert, T. The Application of the Topic Modeling to Question Answer Retrieval. In: Konjović, Z., Zdravković, M., Trajanović, M. (Eds.) ICIST 2016 Proceedings Vol.1, pp.241-246, 2016.

[6] Higgins, D. (2005). Which Statistics Reflect Semantics? Rethinking Synonymy and Word Similarity. In: Kepser, S., Reis, M. (eds.) Linguistic Evidence: Empirical, Theoretical and Computational Perspectives. Mouton de Gruyter, Berlin, pp. 265–284.

[7] Landauer, T et al. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge University of Colorado, Boulder

[8] https://we1s.ucsb.edu/

My Contribution:

Task	Nikita	Sunit	Zhenghao Zhao	Sukhada
Task 1	Imported data for LDA. Analysed preprocessing of respective Data.	Collected data and analysed the data and pre-processed it.	Analysed the data.	Imported data Toefl.Analysed preprocessing of respective Data.
Task 2	Worked on topic-term matrix using topic-state.gz matrix.	Normalized the topic-term matrix	Created topic-term matrix using topic-state.gz matrix.	Analysed the columns in topic-state.gz file that can be used in topic-term matrix.
Task 3	Calculated similarity measure Euclidean distance.	Calculated similarity matrix cosine similarity.	Calculated similarity matrix using cosine distance	Calculated similarity matrix using jensen-shannon divergence.
Task 4	Understanding and analysing concept of word embedding, data and algorithm of word embedding. Worked on creation of matrix for word embedding.	Created the matrix for word embedding. Stemming of TOEFL questions and answers for word embeddings approach.	Created the matrix for word embedding and use it on GRE questions and toefl questions.	Understanding the concept of word embeddings. Worked on stemming of TOEFL questions and answers for word embeddings approach.
Task 5	Created the results in the output file and work on formatting of results. Also did error analysis.	Added the results in the output file. Also did error analysis.	Separated questions according to which part of speech the question word is and did the error analysis	Added questions and its correct answer in the output file. Also did error analysis.