Question Answering System on Global Politics using NLP techniques

Shilpu Srivastava
School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, Ontario, Canada
ssriv071@uottawa.ca

Raunak Mahesh
School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, Ontario, Canada
rmahe005@uottawa.ca

Abstract

A Question-Answering system backed up by a comprehensive knowledge base that helps the user to find the most relevant answers to his queries seems quite desirable in today's world of inquisitiveness. Our work primarily involves building a Question-Answering system from scratch. The scope of this project ranges from building our own corpus by obtaining data from random Wikipedia [1] web pages to querying and retrieving answers based on these webpages. In this paper, we plan to implement a QA system using distinct approaches and compare their results and performance. Our work mainly focuses on exploring Question-Answering using four systems - Closed Domain Question Answering (cdQA), Latent Dirichlet Allocation (LDA), Unsupervised modeling using Sentence Embeddings, and Hierarchical Dirichlet Process (HDP). In normal human-to-human conversations, question-answering seems quite effortless as people are able to comprehend the topic on which the query is based. Likewise, our work on the LDA and HDP predictive modeling emphasizes understanding the topics present across the training dataset prior to answering questions based on it. Additionally, considering the significance of domain specificity in a QA system, we also plan to explore the cdQA based on pretrained SQuAD using BERT [2] and the Unsupervised learning approach of Question-Answering based on Sentence Embeddings. For evaluation, we manually built our test dataset and annotated it with their corresponding expected answers. Our models exhibited an accuracy of 0.62 and 0.77 for the LDA and the Unsupervised Learning approach respectively.

Keywords - Latent Dirichlet Allocation (LDA), Wikipedia, Natural Language Processing (NLP), Part-of-Speech (PoS) tagging, Hierarchical Dirichlet Process (HDP), closed-domain Question Answering (cdQA) architecture, BERT reader, SQuAD 1.1 dataset, Comma Separated Value (CSV), Cosine Similarity, InferSent, Facebook Sentence Embedding, Unsupervised modeling

I. Introduction

A detailed comprehension of the topics on which the questions are based is the key to provide relevant answers. Our question-answering system too has been built with a huge amount of emphasis on understanding the topics present across the entire training dataset. The scope of the work in this project involves building our very own training and testing datasets, performing standard natural language processing (NLP) tasks such as tokenization, lemmatization and part-of-speech (PoS) tagging besides building data models using Latent Dirichlet Allocation (LDA), unsupervised approach and Hierarchical Dirichlet Process (HDP) before finally, answering questions through predictive modelling. The subsequent paragraphs provide an overview of how the dataset was prepared, the approaches that were incorporated while designing the question answering system, the functioning of the system and the motivation for our work.

As in most natural language processing projects, our initial task too was the preparation of the dataset. A choice had to be made to determine if our question answering system was to be an open-domain or a closed-domain system. The fact that a closed-domain question answering system can exploit domain-specific knowledge by using a model that is fitted to a unique-domain dataset [2] justified its selection. It was also decided that the contents pertaining to the global political domain that is freely available on the internet would be incorporated into our dataset. Through our Python code, a random collection of Wikipedia web pages of politicians, actor-turned-politicians, and sportsmen-turned-politicians was accumulated which was then meticulously processed to form the comma-separated value (CSV) dataset. This training dataset included five features or attributes namely - the date of collection of the data, the title of the Wikipedia web page, the link to the Wikipedia web page, the abstract of the Wikipedia web page (which is the opening sentence of the web page) and the paragraphs that existed in that Wikipedia web page.

With the training dataset prepared, comprehending the domain-specific information present in these random Wikipedia web pages by building domain-specific data models on the entire training dataset constituted the second phase of our project. The choice of the approaches that were to be used for building these data models had to be made and it was narrowed down that Latent Dirichlet Allocation (LDA), Hierarchical Dirichlet Process (HDP) and an unsupervised learning approach would be used to build the domain-specific data models. The use of Latent Dirichlet Allocation (LDA) was obvious given that the text present in these random Wikipedia web pages (which can be regarded as web-based 'documents') had to be classified or modelled under more specific topics rather than the very generic 'political' domain. The justification for the choosing Hierarchical Dirichlet Process (HDP) as one of the approaches for model building is that it is a powerful mixed-membership approach for the unsupervised analysis of grouped data [3]. The prominence of an unsupervised learning approach in building a domain-specific data model is of the highest order when the dataset under consideration is an epitome of an unsupervised learning task. Each of these approaches can be looked at as our alternative to the state-of-the-art cdQA [2] model that has been pretrained on the comprehensive, exhaustive SQuAD 1.1 dataset using the BERT reader. Though we have tried to emulate the predictability of the state-of-the-art [2] model through the implementation of our aforementioned alternatives, this should not be mistakenly interpreted that we desire or claim to surpass the state-of-the-art BERT reader [2] model pretrained on SQuAD in its accuracy too.

The input to the closed domain question answering system is a simple, domain-specific (global politics) question such as - Name a Canadian politician and the seventh Lieutenant Governor of Nova Scotia. Upon receiving this test input, the system extracts domain-specific knowledge from the models constructed using the three, aforementioned approaches. In other words, each of the three models developed using Latent Dirichlet Allocation (LDA), Hierarchical Dirichlet Process (HDP) and an unsupervised learning approach predict their answers respectively to the inputted question. Each of these predicted answers are then evaluated against the expected answer (Sir Malachy Bowes Daly) using cosine similarity. This similarity evaluation has been made possible by a finite set of test data (questions) which consists of the question and it's expected, correct answer. This test data has been meticulously prepared considering the various test-case scenarios. The predicted answer that is most similar to the expected answer is then rendered as the output (answer).

The motivation for this work is instilled from the primary objective of the project which is to utilize the concepts covered during in class lectures besides learning numerous other natural language processing concepts and to build on these concepts whilst implementing the question-answering system. The scope of our work has enabled us to achieve this objective by facilitating the learning of core natural language processing concepts such as tokenization, lemmatization and part-of-speech (PoS) tagging besides other trending concepts such as Latent Dirichlet Allocation (LDA), Hierarchical Dirichlet Process (HDP) and closed-domain Question Answering (cdQA) architecture. Another motivation to take up this project, is the significant application of the such a Question Answering System in the real world. Our system can be useful in many practical applications like Quizzing Platforms, Examination and Academic Evaluation purposes, it also can be widely used in the Information Retrieval domain.

II. Literature Review

Celikyilmaz et al.,(2010) present an exploration of generative modeling for the question answering task to rank candidate passages. [9] Their paper presents approaches on similarity and discriminative modeling. Similarity modeling has been achieved using Latent Dirichlet Allocation (LDA) and Hierarchical Latent Dirichlet Allocation (hLDA) models. The LDA model has been constructed to calculate the degree of similarity between each question-answer pair based on two measures. Gibbs sampling has been utilized to fit the hLDA model which has been built to discover hidden topic distributions for a given question. Discriminative modeling has been implemented by extending their baseline question answering model (supervised classifier model) to evaluate three different models - Models M-1, M-2 and M-3 [9]. The results of their experiments suggest that extracting information from hidden concepts improves the results of a classifier-based QA model [9]. This suggestion has been incorporated into our work in a bid to enhance the classification results of our system.

The study of Louvigné et al.,(2013) analyzes a corpus of 1,500 international conference papers of the International Conference on Advanced Learning Technologies (ICALT) using corpus linguistic tool and applies the LDA (Latent Dirichlet Allocation) algorithm [10]. The fact that most technical authors make use of their own linguistic patterns in their writings is at the center of their study. Their work is aimed at disinterring such hidden linguistic patterns present in the abstracts of numerous international conference papers using six "rhetorical moves" [10]. The implementation involves the use of AntConc (a corpus linguistics tool) [10] and the Latent Dirichlet Allocation (LDA) algorithm for semantic analysis. The list of topics (research areas) present within the corpus of abstracts has been extracted using the LDA algorithm. Likewise, LDA has been incorporated in our work so as to extract the numerous topics that were present in our closed-domain dataset.

Anupriya and Karpagavalli.,(2015) [11] conducted a detailed, comparative study of Latent Dirichlet Allocation (LDA) estimation with Collapsed Variational Bayesian (CVB) and Gibbs sampling. The comparative study was performed on a set of two hundred abstracts belonging to various research papers in the field of computer science and medicine. ScalaNLP [12] was used to pre-process and clean the contents of these abstracts. The experiment was conducted by providing the same value of 0.01 for both the fixed hyper-parameters (and) [11]. Perplexity is used as an evaluation measure to assess the strength of built models. The results showed that the extracted topics capture meaningful structure in the data [11] besides, reiterating the fact that a large collection of documents can be subjected to unsupervised analysis using topic modeling. It is with this rationale that we have included topic modeling on our collection of random Wikipedia web pages (documents) so as to capture meaningful structures in the data through the extracted topics.

Duan et al.,(2015) [14] introduce a recommendation system for microblogs that has been designed using the Latent Dirichlet Allocation (LDA) model. The contents of the microblog are classified into various topics with the help of the LDA model. The authors accumulated about 600,000 microblogs as documents between May 2014 to November 2014 [14]. The information present in these documents was modelled into numerous topics such as sports, science, current events, people's livelihood, entertainment and other categories [14] using the LDA model. Based on the user's interests, these topics were further amalgamated into clusters of similar topics in an attempt to avoid repeated recommendation to the user. The authors used perplexity as an evaluation parameter to measure the performance of the constructed LDA model. After reviewing this literature, we were left induced to consider perplexity as an evaluation criteria in our work.

Troussas et al.,(2017) [15] present a system wherein the Latent Dirichlet Allocation (LDA) model bolsters the learning of contexts through continuous observation of the user's activity and interests on social networking platforms such as Facebook. Learning contexts enables the system to make automatic predictions through the LDA model by using probabilistic approaches. Interestingly, the utilization of LDA models to make predictions based on a user's social media activities and interests is an innovative design decision. Thus, in an attempt to emulate this design decision we decided to use the contents of Wikipedia pages to automatically predict answers using LDA models in our work. The future work of the literature provides a suggestion to evaluate the usefulness of such automatic predictions which again has been incorporated into the evaluation of the results obtained through our work.

III. Methodology

The literature we reviewed as part of this project, provided us an insight into various ideas that have been used in the past to build such QA systems. We tried to explore different NLP solutions that would help us design such a system and give us results comparable to some of the accurate systems that have been developed already. The Stanford Question Answering Dataset(SQuAD) is one of the most accurate models that we could come across. It is a reading comprehension dataset, with around 100000+ question-answer pairs on more than 500 Wikipedia articles. This dataset is significantly huge and has given promising results in training various question answering system.

We used the SQuAD dataset as a benchmark for two areas in our system. Firstly, we wanted to build our QA system also at a reading comprehension level, hence the corpus that we chose for our project was also built across 1500+ Wikipedia pages on some of the famous politicians, actor-politicians, and sportsmen politicians. Secondly, the SQuAD dataset modeling has seen to give very high accuracy for the answers and we would like to consider the same as the state-of-the-art model for building our Question-Answering system. Our system, therefore, is a closed domain Question Answering system where the domain corresponds to famous politicians, actor-politicians, and sportsmen politicians.

We explored four distinct methods for the purpose of our implementation: the CDQA pipeline approach, the LDA approach, the HDP approach, and an Unsupervised approach based on Facebook Sentence Embedding. We would like to provide a brief overview of each of these methods in the subsections to follow

A. *cdQA Approach*

The cdQA approach [2] is the implementation of a closed domain question answering system. The cdQA test suite provides a Python package that enables the user to implement a cdQA pipeline that enables closed domain question answering on the corpus. The cdQA architecture comprises of two major components: the Retriever and the Reader. Here, the Retriever is trained on a pool of articles using a tf-idf matrix. Whenever a question arises, it's provided as an input to the Retriever in the form of a tf-idf vector. The Retriever calculates the similarity scores and fetches the documents having the most similarity with the query. The paragraphs of the most similar documents are then fed into the Reader, which in turn outputs the results. Figure 1 shows the architecture of the cdQA system. We used this cdQA pipeline architecture to query our dataset based on a pretrained BERT model which has been trained on the SQuAD dataset.

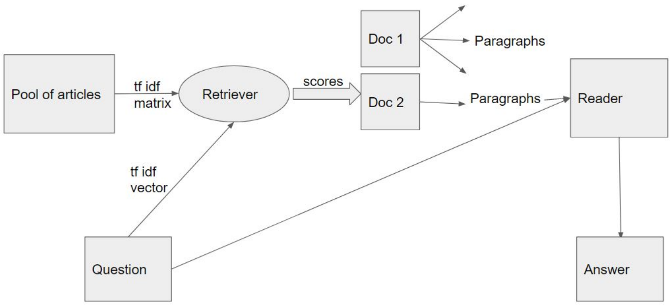


Fig. 1. Adapted from Farias et al. [2]

B. *Latent Dirichlet Analysis Approach*

The Latent Dirichlet Allocation method is used to detect abstract topics across a list of documents. We perform LDA modeling on our corpus to extract the abstract topics that define most of the sentences in the data. Using some parameters like the coherence score and model perplexity, we determine the optimal number of topics and learning rates and train the LDA model. After determining the topics, we try to predict topics on the unseen document (in our case, question) and extract the most relevant document that might probably consider the answer. After extracting the document, we use some spaCy libraries like Pattern Matcher based on POS tagging and EntityRuler to extract the exact answer from the predicted document.

C. *Hierarchical Dirichlet Allocation*

We also explored the Hierarchical Dirichlet Allocation approach. The HDP model is a fully unsupervised model which determines its optimum number of topics using posterior interference. The HDP models topics as a combination of words quite similar to how LDA works. The difference between the LDA and HDP model is that the LDA model classifies a document across a fixed number of topics while the HDP the number of topics is determined by a Dirichlet process which makes the topic a random variable as well [4]. As the name suggests, "hierarchical" the HDP model is an extension of the LDA model.

D. *Unsupervised modeling with InferSent (Facebook Sentence embedding)*

Our unsupervised approach involves the usage of the Facebook Sentence Embeddings to transform the sentences to their corresponding vectors. The Facebook sentence embedding is named as Infersent [5]. The other traditional embeddings like word2vec, doc2vec, node2vec, represent the entities using vectors of various dimensions to represent the string as float values. Such a conversion of string to numerical values, enables the computer to understand the semantic representations better. The traditional approach is to tokenize a sentence into words, convert them into vectors using GloVe embedding and take the average of these vectors to convert them into a bag of words. This approach fairs decently but does not accurately consider the order of the words to determine the exact sequence of the sentence.

Facebook's Infersent library is a sentence embedding that provides semantic representation for English sentences. It is trained on the Natural Language Interface data. The Infersent models have two versions, one trained on the GloVe embedding and the other trained on fastText. We downloaded the Infersent model from the github repository and set the GloVe and fastText word-vector paths. Then we built the vocabulary of word vectors and encoded the sentences.

After converting the sentences to vectors, we used an unsupervised approach [6] where we used the cosine similarity to relate the questions to the answers. In order to account for the similarity, cosine similarity between the sentence vectors seems to be more effective than the Euclidean distance [6] because the cosine similarity takes into account the angle between the vectors as well. On the basis of cosine similarity, we identify the most similar abstract as the predicted abstract that might potentially consider the answer. After getting the predicted paragraph, we use some spaCy libraries like Pattern Matcher based on POS tagging and EntityRuler to extract the exact answer from the predicted document.

IV. Building the Corpus

As with most natural language processing projects, we had to gather and prepare our own data for the project. The fact that we had to collect the data specific to a particular domain (closed-domain) narrowed down our quest for the data. Our interest in geopolitics influenced the choice of global politics as the project's domain. The fact that information pertaining to the global political domain is easily accessible and freely available on the internet further reinforced the choice of domain. With the domain selected, our focus shifted towards determining the source from where the domain-specific information could be retrieved. At this juncture it was decided that we would obtain the necessary information directly from the web pages on the internet rather than scouting around the internet for existing, structured documents. This step can be regarded as the most imperative phase of the project as the models that shall be constructed can completely exploit domain-specific knowledge only if this information is relevant and sufficient.

The names of numerous politicians, actor-turned-politicians and sportsmen-turned-politicians from across the globe were randomly, manually collected and their corresponding Wikipedia web pages were retrieved. The finite collection of 1,605 Wikipedia web pages of global politicians was meticulously processed with the help of our Python code to form a comma separated value (CSV) dataset with five attributes or features. The five attributes are 'date', 'title', 'link', 'abstract' and 'paragraphs' which contain the date of collection of the data, the title of the Wikipedia web page, the link to the Wikipedia web page, the abstract of the Wikipedia web page (which is the opening sentence of the web page) and the paragraphs that existed in that Wikipedia web page. Thus, a dataset of 1,605 x 5 dimensionality was prepared as a prerequisite to the build domain-specific data models for our closed-domain question-answering system.

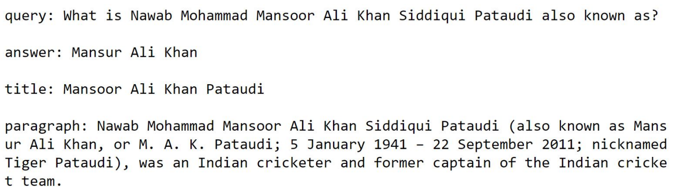
V. Design and Implementation

We tried to explore the four methods listed further in this section in order to build a Question Answering system that could retrieve the most accurate answer from our corpus on politicians, sports politicians and actor-turned politicians. In order to query our corpus, we manually created a set of 400 questions. This set consists of below types of queries based on various randomly selected politicians from our corpus:

- Name a person who ... (Answer) X
- Who is X?
- What is X also known as?
- When was X born?
- When did X die?

A. *cdQA*

For implementing the cdQA architecture, we used the cdqa.utils and the cdqa.pipeline library which is a python package to implement the QA pipeline. The cdQA utils library enables us to download any pre-trained model and use the same as the training data for the QA pipeline [2]. We downloaded the pre-trained BERT model built on the SQuAD 1.1 dataset and provided it as the training data to the QA pipeline. Then we loaded our corpus into a dataframe. The cdQA pipeline allows us to fit into it's retriever our own corpus. After fitting the corpus to the cdQA's retriever, we simply need to pass our query into the pipeline into the predict() function, that gives back the answer to the query. The figure 2 below shows a sample of the results obtained from the cdQA model when we provided a query built from our own corpus.



```
query: What is Nawab Mohammad Mansoor Ali Khan Siddiqui Pataudi also known as?

answer: Mansur Ali Khan

title: Mansoor Ali Khan Pataudi

paragraph: Nawab Mohammad Mansoor Ali Khan Siddiqui Pataudi (also known as Mans
ur Ali Khan, or M. A. K. Pataudi; 5 January 1941 – 22 September 2011; nicknamed
Tiger Pataudi), was an Indian cricketer and former captain of the Indian cricke
t team.
```

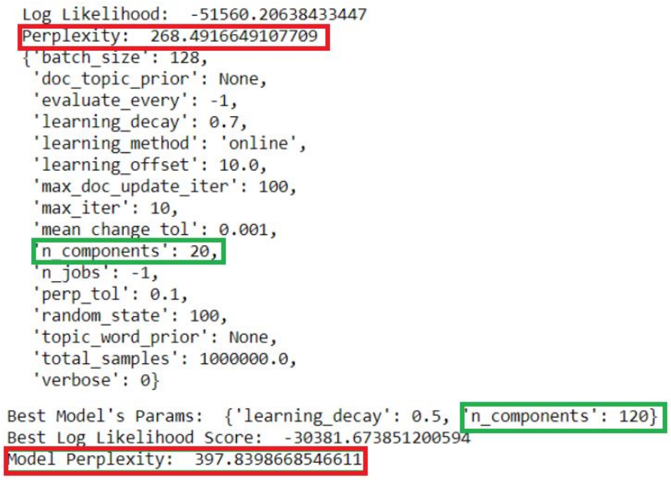
Fig. 2. cdQA Question-Answer output on Personalities Corpus

Since the training dataset that we used for the cdQA architecture was a BERT model built on the SQuAD 1.1 version, the predictions made by the system were highly accurate. We therefore considered the results obtained by the cdQA pipeline pretrained on the SQuAD dataset as *state-of-the-art* and wanted to explore some other methods where we could actually perform training on our own corpus and achieve comparable results.

B. *QA using Latent Dirichlet Allocation*

For training our dataset based on the Latent Dirichlet Allocation Method, we used sklearn as well as gensim libraries. We used the LatentDirichletAllocation, TruncatedSVD libraries from sklearn.decomposition to implement this method. We used the LDAMallet model from the gensim library as well to train our model and determine the optimal parameters like the number of topics and learning rate that could enhance the coherence scores of the model and reduce the perplexity of the system. Illustrated below is the step-by-step process that we followed to implement this system. Firstly, we cleaned our dataset using regex, where we removed the disturbing special characters which were not very useful in our corpus. Since, we built the corpus on our own using 1500+ wikipedia pages, our corpus had a lot of special characters that required cleaning. We used the gensim.simple_preprocess [7] library to preprocess the sentences. Using this library, we removed punctuations and also returned the tokens for each sentence. We removed the stop-words from the corpus. Then we lemmatized the sentences using the spaCy “en_core_web_sm” library. We only allowed the POS tags of [’NOUN’, ’ADJ’, ’VERB’, ’ADV’] for the lemmatization of our data. After lemmatizing our data, we converted the lemmatized data to vectors using the CountVectorizer library from sklearn. We also materialized the sparse data and the sparsity of our data was observed to be around 0.77Next we built our LDA model. To build our LDA model, we tried both the sklearn implementation as well as the gensim implementation. And both these models roughly suggested us the same number of topics as the optimal hyperparameters.

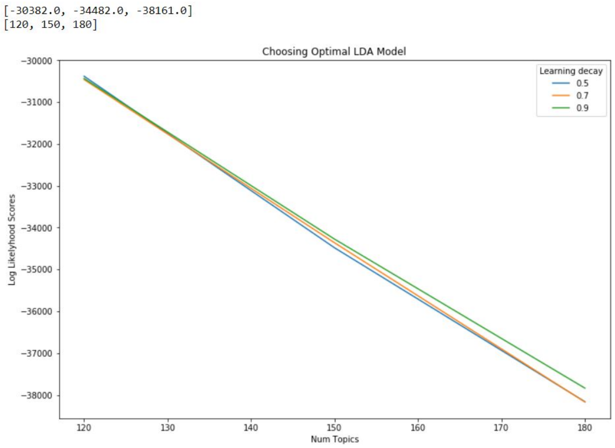
*1) **LDA using Sklearn**:* We initially trained the LDA model using the number of topics as 20. The evaluation parameters for the sklearn library are perplexity and log likelihood ratio. We initially saw the perplexity as 270.60 and we could observe that the perplexity was increasing on increasing the number of topics(we could see a perplexity of 486.88 for number of topics = 150). However, this behavior is not expected ideally, there seems to be an error in the sklearn’s implementation [8]. We also tried to perform a grid search on the number of topics to search for the most optimal parameters. But we could see that the log likelihood ratio continuously decreased with the number of topics. Figure 3 and 4 shows the results of the grid search. Since we were not very convinced with the results of sklearn’s hyperparameters tuning, we decided to check the same on the LDA model using the gensim library.



```
 Log Likelihood:  -51560.20638433447
 Perplexity:  268.4916649107709
 {'batch_size': 128,
  'doc_topic_prior': None,
  'evaluate_every': -1,
  'learning_decay': 0.7,
  'learning_method': 'online',
  'learning_offset': 10.0,
  'max_doc_update_iter': 100,
  'max_iter': 10,
  'mean_change_tol': 0.001,
  'n_components': 20,
  'n_jobs': -1,
  'perp_tol': 0.1,
  'random_state': 100,
  'topic_word_prior': None,
  'total_samples': 1000000.0,
  'verbose': 0}

Best Model's Params:  {'learning_decay': 0.5, 'n_components': 120}
Best Log Likelihood Score:  -30381.673851200594
Model Perplexity:  397.8398668546611
```

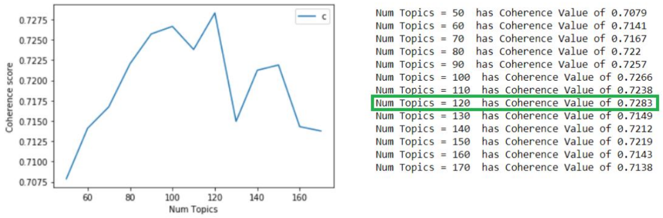
Fig. 3. Model Perplexity Scores reported by Sklearn’s LDA model



```
[-30382.0, -34482.0, -38161.0]
[120, 150, 180]
```

Fig. 4. Model Log Likelihood Scores reported by Sklearn’s LDA model

*2) **LDA using Gensim**:* In the gensim library, the training evaluation parameters are model perplexity and coherence scores. We started with the number of topics as 20 and the model reported a perplexity of -19.02 and a coherence score of 0.505. Next we decided to plot a graph between the coherence scores of the number of topics between 50 and 180. We accomplished this using the gensim.models.wrappers.LdaMallet library. The figure 5 shows the coherence scores of the models. At the number of topics as 120, the coherence was observed to be the highest, the value being 0.7283.

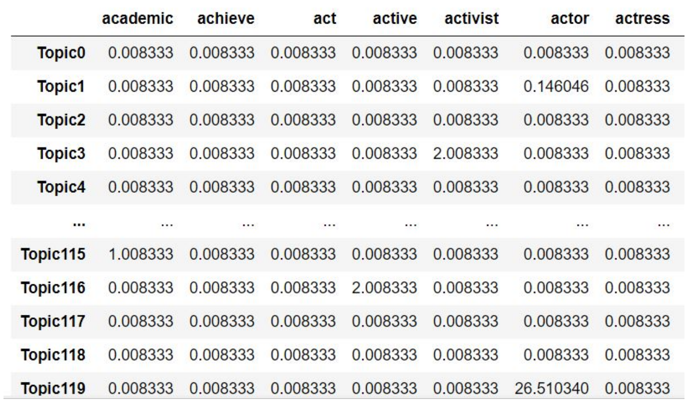


```
Num Topics = 50  has Coherence Value of 0.7079
Num Topics = 60  has Coherence Value of 0.7141
Num Topics = 70  has Coherence Value of 0.7167
Num Topics = 80  has Coherence Value of 0.722
Num Topics = 90  has Coherence Value of 0.7257
Num Topics = 100  has Coherence Value of 0.7266
Num Topics = 110  has Coherence Value of 0.7238
Num Topics = 120  has Coherence Value of 0.7283
Num Topics = 130  has Coherence Value of 0.7149
Num Topics = 140  has Coherence Value of 0.7212
Num Topics = 150  has Coherence Value of 0.7219
Num Topics = 160  has Coherence Value of 0.7143
Num Topics = 170  has Coherence Value of 0.7138
```

Fig. 5. Coherence scores reported by Gensim's LDAMallet Model.

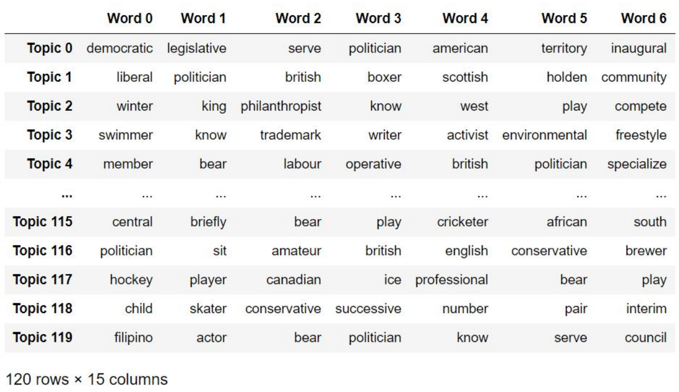
With the increase in the number of topics from 20 to 120, the model perplexity decreased from -19.02 to -238.27 and the model coherence increased from 0.505 to 0.7283. Hence, we decided to choose our optimal number of topics as 120 as it gave us the most optimal results in terms of model perplexity and model coherence.

Then we visualized our LDA model built. Figure 5 shows features (word vectors) across each topic. Figure 6 shows the words distributed across each of the 120 topics.



	academic	achieve	act	active	activist	actor	actress
Topic0	0.008333	0.008333	0.008333	0.008333	0.008333	0.008333	0.008333
Topic1	0.008333	0.008333	0.008333	0.008333	0.008333	0.146046	0.008333
Topic2	0.008333	0.008333	0.008333	0.008333	0.008333	0.008333	0.008333
Topic3	0.008333	0.008333	0.008333	0.008333	2.008333	0.008333	0.008333
Topic4	0.008333	0.008333	0.008333	0.008333	0.008333	0.008333	0.008333
...	...	...	...	...	...	...	...
Topic115	1.008333	0.008333	0.008333	0.008333	0.008333	0.008333	0.008333
Topic116	0.008333	0.008333	0.008333	2.008333	0.008333	0.008333	0.008333
Topic117	0.008333	0.008333	0.008333	0.008333	0.008333	0.008333	0.008333
Topic118	0.008333	0.008333	0.008333	0.008333	0.008333	0.008333	0.008333
Topic119	0.008333	0.008333	0.008333	0.008333	0.008333	26.510340	0.008333

Fig. 6. Word Vectors across 120 Topics



	Word 0	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6
Topic 0	democratic	legislative	serve	politician	american	territory	inaugural
Topic 1	liberal	politician	british	boxer	scottish	holden	community
Topic 2	winter	king	philanthropist	know	west	play	compete
Topic 3	swimmer	know	trademark	writer	activist	environmental	freestyle
Topic 4	member	bear	labour	operative	british	politician	specialize
...	...	...	...	...	...	...	...
Topic 115	central	briefly	bear	play	cricketer	african	south
Topic 116	politician	sit	amateur	british	english	conservative	brewer
Topic 117	hockey	player	canadian	ice	professional	bear	play
Topic 118	child	skater	conservative	successive	number	pair	interim
Topic 119	filipino	actor	bear	politician	know	serve	council

120 rows × 15 columns

Fig. 7. Words distributed across 120 Topics

We also visualized the clusters using K-Means clustering. For this we first performed a dimensionality reduction on the lda model where we used the Singular Value Decomposition(SVD) to transform the LDA model to 2 dimensions.We gave the number of clusters as 15. Figure 8 shows the segregation of clusters as obtained with the k-Means clustering with SVD transformation.

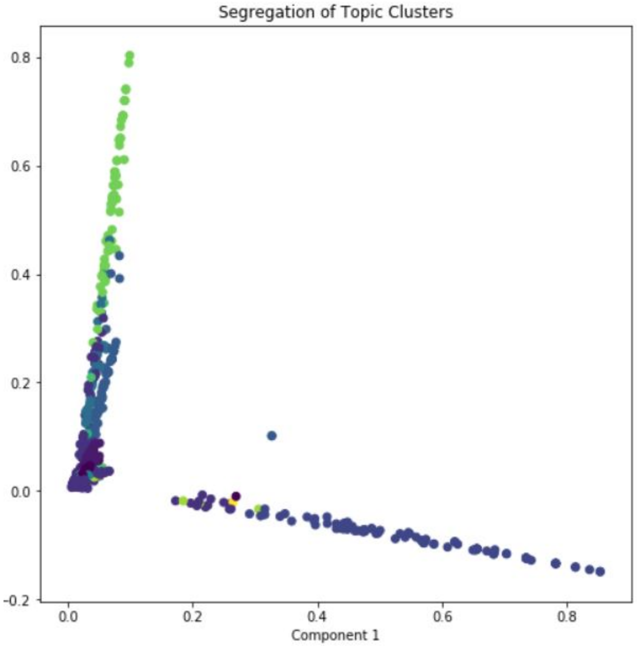
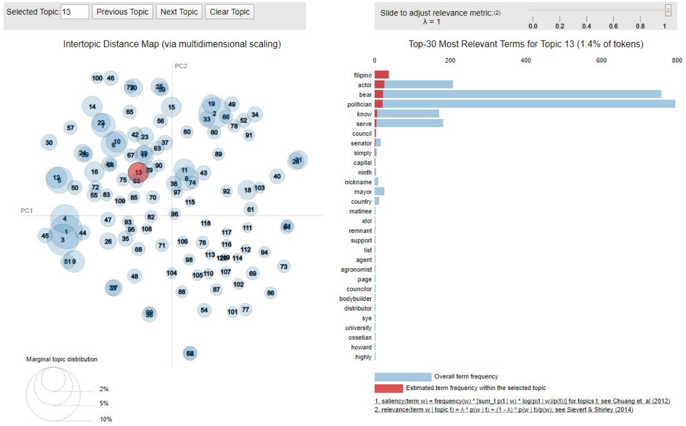


Fig. 8. Segregation of Topic Clusters using k-Means Clustering

We used the pyLDAvis library to visualize the intertopic distance map and the top 30 most relevant terms for each of the 120 topics. Figure 9 illustrates the same. We can see how words are shown across the topics based on the inter-topic distance.



Selected Topic: 13

Slide to adjust relevance metric:(2)
λ = 1

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

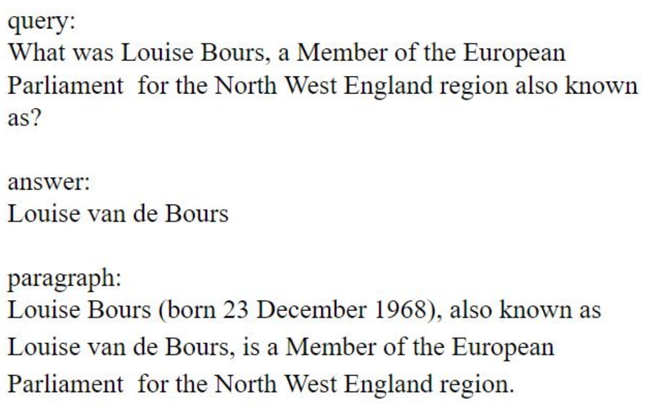
Fig. 9. pyLDAvis Visualization of the LDA model

After visualizing our LDAModel, we decided to predict our model on an unseen document(in our case, the user's query). As soon as the user enters a query, it is preprocessed in the same way as our corpus was preprocessed. We apply gensim.simple_preprocessing, stopwords removal and lemmatization on the query. Then we convert the query's tokens to a vector using the CountVectorizer. After this preprocessing, we fit it into our LDA model using LDA transform and which gives us the topic and its corresponding topic probability scores. We have implemented a function that uses the euclidean distance to fetch the most similar documents on the basis of minimum euclidean distance. The most relevant document (in our case, the paragraph/abstract) is retrieved using this function.

In order to get the exact answer from the most relevant paragraph, we further perform text processing spaCy Matcher and EntityRuler. We perform rule based matching using the Entity Ruler. Entity Ruler is a way to match using a dictionary containing the key-value pairs. It consists of a "label" and a "pattern" [13]. When a pattern match is observed, the label is returned.

We defined various patterns for the typical questions that could be asked on our corpus. If we talk about a person X defined in our corpus, some of the questions that could be asked are: "Who is X?", "When was X born?" , "What is X also known as?", "When did X die?". In the EntityRuler, we defined patterns for each of these types of questions. For instance, for "aka" we defined accepted patterns as phrases like "also known as", "'nicknamed", "known mononymously as", "known professionally as". We also perform POS tagging on the output to ensure that the results correspond to what is expected. For example, the query "also known as" is expected to return the result as a PROPN/NN (Noun or Proper Noun only), hence our matcher also checks for the POS tagging of the results obtained.

Figure 10 shows the results of a query on our LDA QA system.


query:
What was Louise Bours, a Member of the European Parliament for the North West England region also known as?

answer:
Louise van de Bours

paragraph:
Louise Bours (born 23 December 1968), also known as Louise van de Bours, is a Member of the European Parliament for the North West England region.

Fig. 10. Output obtained from QA system built using LDA

C. *QA using Hierarchical Dirichlet Processing*

The implementation of the Hierarchical Dirichlet Process model was carried out using the gensim library. We preprocessed our corpus using gensim.simple_process, we removed the stopwords and extracted the bigrams. These bigrams were further lemmatized to include only allowed POS of 'NOUN', 'ADJ', 'VERB', 'ADV' (Nouns, Adjectives, Verbs, Adverbs). After lemmatizing the text, we prepared our corpus by passing it into a gensim Corpora dictionary (corpora.Dictionary) and then converting finally the lemmatized data into a bag of words by using the "doc2bow" embedding. We then pass this corpus and our corpora dictionary to the HDP model. We calculated the length of the alpha value and got to know that the HDP model chose the best value of the number of topics as 150. The model gave a perplexity of -167.585 and a coherence score of 0.77.

The figure 11 below shows some words classified across the topics by the HDP model.

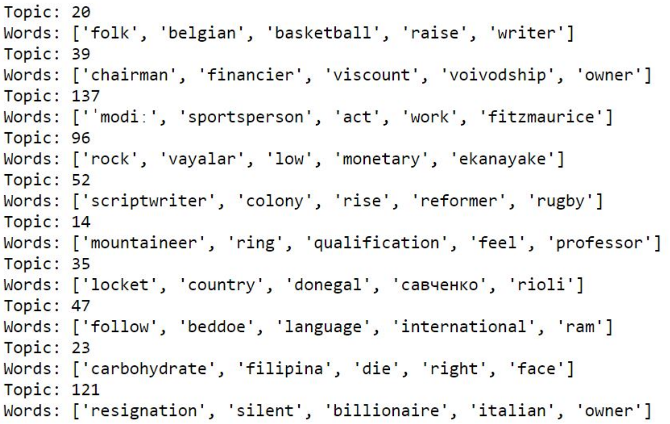
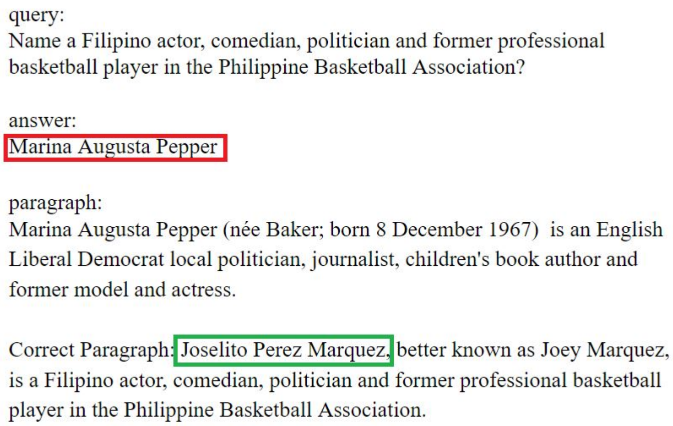

```
Topic: 20
Words: ['folk', 'belgian', 'basketball', 'raise', 'writer']
Topic: 39
Words: ['chairman', 'financier', 'viscount', 'voivodship', 'owner']
Topic: 137
Words: ['ˈmodiː', 'sportsperson', 'act', 'work', 'fitzmaurice']
Topic: 96
Words: ['rock', 'vayalar', 'low', 'monetary', 'ekanayake']
Topic: 52
Words: ['scriptwriter', 'colony', 'rise', 'reformer', 'rugby']
Topic: 14
Words: ['mountaineer', 'ring', 'qualification', 'feel', 'professor']
Topic: 35
Words: ['locket', 'country', 'donegal', 'савченко', 'rioli']
Topic: 47
Words: ['follow', 'beddoe', 'language', 'international', 'ram']
Topic: 23
Words: ['carbohydrate', 'filipina', 'die', 'right', 'face']
Topic: 121
Words: ['resignation', 'silent', 'billionaire', 'italian', 'owner']
```

Fig. 11. Top 5 Words classified across topics using HDP

However, we could observe that we could not get accurate results from the HDP model. For the question in Figure 12, the answer obtained by the model seems to be incorrect. The document fetched by the HDP model (shown in red) was different to the expected document for the question (shown in green).


query:
Name a Filipino actor, comedian, politician and former professional basketball player in the Philippine Basketball Association?

answer:
Marina Augusta Pepper

paragraph:
Marina Augusta Pepper (née Baker; born 8 December 1967) is an English Liberal Democrat local politician, journalist, children's book author and former model and actress.

Correct Paragraph: Joselito Perez Marquez, better known as Joey Marquez, is a Filipino actor, comedian, politician and former professional basketball player in the Philippine Basketball Association.

Fig. 12. QA output of HDP model

D. QA using Unsupervised modeling with InferSent

In order to build the QA model using unsupervised modeling, we used InferSent which is the Facebook Sentence Embedding [6] that enables Natural Language Inference. Firstly, we preprocessed out data using TextBlob. We also require NLTK and the “Punct” tokenizer. We downloaded the InferSent library, which helps us to provide semantic sentence representations by converting the sentences to its corresponding vectors [16]. We also download the state-of-the-art fastText and GloVe word embeddings as the InferSent models are trained on these emneddings. We built the vocabulary using Infersent based on our corpus. Where each sentence of our corpus was converted to its corresponding vector using the InferSent.

For example, the below sentence from our corpus was converted to its corresponding sentence vector as below:

Barack Hussein Obama II ((listen); born August 4, 1961) is an American politician and attorney who served as the 44th president of the United States from 2009 to 2017.

[[-0.02874381 0.04943122 0.20910737 ... 0.1316887 0.04436744 0.11170954]]

We also passed the list of 400 questions that we built for testing, to the InferSent model to add it to the vocabulary. All these questions are also converted to their corresponding sentence vectors.

For example, the below user query was converted to its corresponding InferSent Embedding as below:

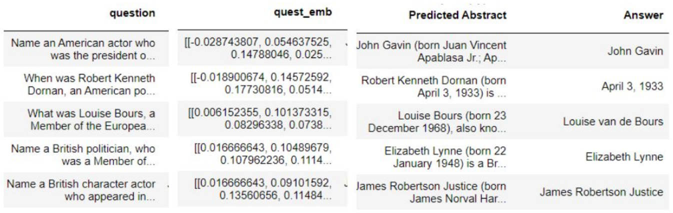
Who was William Spiers ”Billy” Glenn?’

[[-0.01890067, 0.04943122, 0.07359568, ..., 0.03476685, 0.02266138, 0.01821707]]

So now, we had a *dictionary* that consisted of our corpus as well as the manually built set of 400 questions that we would be utilizing to query our corpus.

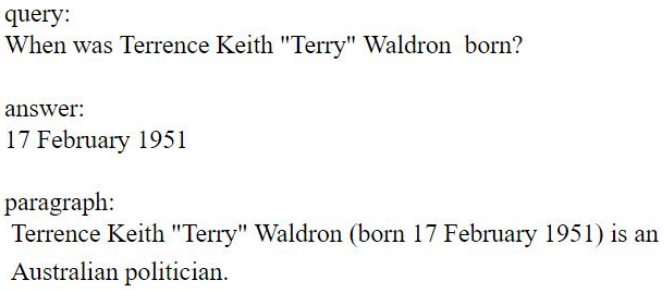
Using this dictionary, we used the cosine similarity to find the most similar document to the question. The cosine similarity was calculated between the sentence vectors of the documents and the user query. After extracting the most similar document, in order to extract the accurate answer from the paragraph, we again made use of the spaCy Matcher and the Entity Ruler. We performed pattern based matching based on the patterns we predefined in the EntityRuler and returned the corresponding label of the matched pattern.

Figures 13 and 14 below show a sample of results obtained from our Unsupervised-InferSent model.



question	quest_emb	Predicted Abstract	Answer
Name an American actor who was the president o...	[[-0.028743807, 0.054637525, 0.14788046, 0.025...	John Gavin (born Juan Vincent Apablasa Jr.; Ap...	John Gavin
When was Robert Kenneth Dornan, an American po...	[[-0.018900674, 0.14572592, 0.17730816, 0.0514...	Robert Kenneth Dornan (born April 3, 1933) is ...	April 3, 1933
What was Louise Bours, a Member of the Europea...	[[0.006152355, 0.101373315, 0.08296338, 0.0738...	Louise Bours (born 23 December 1968), also kno...	Louise van de Bours
Name a British politician, who was a Member of...	[[0.016666643, 0.10489679, 0.107962236, 0.1114...	Elizabeth Lynne (born 22 January 1948) is a Br...	Elizabeth Lynne
Name a British character actor who appeared in...	[[0.016666643, 0.09101592, 0.13560656, 0.11484...	James Robertson Justice (born James Norval Har...	James Robertson Justice

Fig. 13. QA output of the Unsupervised-InferSent model



query:
When was Terrence Keith "Terry" Waldron born?

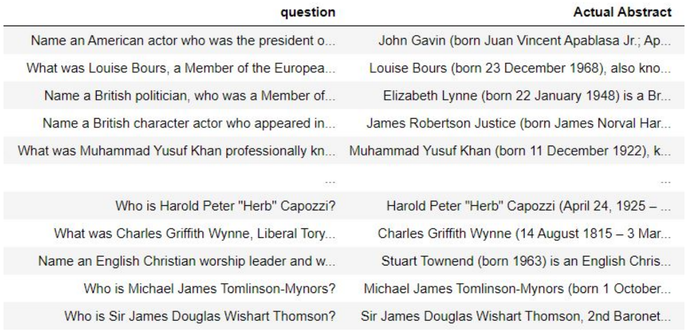
answer:
17 February 1951

paragraph:
Terrence Keith "Terry" Waldron (born 17 February 1951) is an Australian politician.

Fig. 14. QA output of the Unsupervised-InferSent model

VI. Evaluation Strategy

In order to evaluate our results, we manually built a testing dataset with 400 queries that enabled us to evaluate how good our models were performing. We also manually annotated the questions set with their corresponding expected paragraphs where the answer of the question is supposed to be found. Our testing dataset looks like the below:



question	Actual Abstract
Name an American actor who was the president o...	John Gavin (born Juan Vincent Apablasa Jr.; Ap...
What was Louise Bours, a Member of the Europea...	Louise Bours (born 23 December 1968), also kno...
Name a British politician, who was a Member of...	Elizabeth Lynne (born 22 January 1948) is a Br...
Name a British character actor who appeared in...	James Robertson Justice (born James Norval Har...
What was Muhammad Yusuf Khan professionally kn...	Muhammad Yusuf Khan (born 11 December 1922), k...
...	...
Who is Harold Peter "Herb" Capozzi?	Harold Peter "Herb" Capozzi (April 24, 1925 – ...
What was Charles Griffith Wynne, Liberal Tory...	Charles Griffith Wynne (14 August 1815 – 3 Mar...
Name an English Christian worship leader and w...	Stuart Townend (born 1963) is an English Chris...
Who is Michael James Tomlinson-Mynors?	Michael James Tomlinson-Mynors (born 1 October...
Who is Sir James Douglas Wishart Thomson?	Sir James Douglas Wishart Thomson, 2nd Baronet...

Fig. 15. Question Set with Annotated Answer Paragraphs

The questions were built corresponding to most of the politicians, actor-turned-politicians and sportsmen politicians. A sample document(a paragraph from our corpus) and the possible questions that our system can answer are enlisted below:

Paragraph/Abstract:

Patrick Munro (9 October 1883 - 3 May 1942), also known as Pat Munro, was a Scotland international rugby union player and later a British Conservative politician.

Descriptive Questions: *Who is Patrick Munro?*

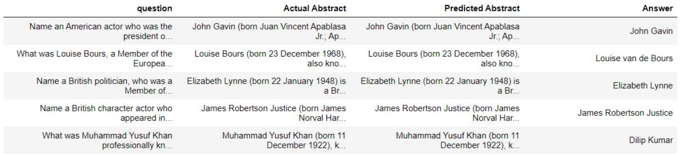
Objective Questions: *Name an international rugby union player and later a British Conservative politician? When was Patrick Munro born? When did Patrick Munro die? What was Patrick Munro also known as?*

The 400 test queries are divided across the types of questions as shown in Table I:

TABLE I
TEST-QUESTIONS USED TO QUERY THE CORPUS

Question Type	Question	No of Questions
Descriptive	Who is X?	100
Objective	Name a politician who...	173
Objective	When was X born?	54
Objective	When did X die?	34
Objective	What is X also known as?	39

The models on predicting, return the Predicted Paragraphs and their actual Answers as shown in Figure 16:



question	Actual Abstract	Predicted Abstract	Answer
Name an American actor who was the president o...	John Gavin (born Juan Vincent Apablasa Jr.; Ap...	John Gavin (born Juan Vincent Apablasa Jr.; Ap...	John Gavin
What was Louise Bours, a Member of the Europea...	Louise Bours (born 23 December 1968), also kno...	Louise Bours (born 23 December 1968), also kno...	Louise van de Bours
Name a British politician, who was a Member of...	Elizabeth Lynne (born 22 January 1948) is a Br...	Elizabeth Lynne (born 22 January 1948) is a Br...	Elizabeth Lynne
Name a British character actor who appeared in...	James Robertson Justice (born James Norval Har...	James Robertson Justice (born James Norval Har...	James Robertson Justice
What was Muhammad Yusuf Khan professionally kn...	Muhammad Yusuf Khan (born 11 December 1922), k...	Muhammad Yusuf Khan (born 11 December 1922), k...	Dilip Kumar

Fig. 16. Output obtained from Test questions

In order to evaluate the accuracy of the answers being predicted by our model, we wanted to analyse the similarity between the actual and predicted paragraphs. Hence, we calculated the *cosine similarity* between the actual paragraph and the predicted paragraph for each of the 400 questions in our testing dataset.

VII. RESULTS ANALYIS

We analysed the results of the below three models based on the cosine similarity of the predicted paragraph (paragraph predicted by the model) and the actual paragraph(manually annotated paragraph containing the expected answer) for each of the 400 queries. The table II below shows the accuracy (based on cosine similarity) and the time taken to predict the answers:

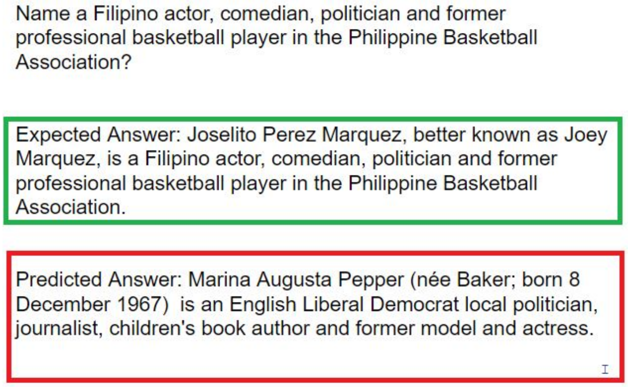
TABLE II
RESULT ANALYSIS OF QA MODELS

Model	Accuracy	Answer Time(400 queries)
cdQA pretrained on SQuAD	0.9342	2.5 hours
LDA	0.6195	less than 5 mins
Unsupervised-InferSent	0.7730	less than 5 mins

As shown in Table II, we can see that in terms of accuracy the cdQA model showed the highest accuracy of 93%, the Unsupervised-InferSent model showed an accuracy of 77%, while the Latent Dirichlet Allocation model showed an accuracy of 62%.

However, we observed that though the cdQA model pre-trained on SQuAD gave highly accurate results, it took a very long time (approximately 2.5 hours) to predict the answers. While the other two models completed their training as well as prediction in less than 5 minutes.

We observed that the HDP model was not performing very well in terms of making predictions. Figure 17 shows a sample answer we received from the HDP model, where the predicted answer(red) seems to be very different from the expected answer(green).



Name a Filipino actor, comedian, politician and former professional basketball player in the Philippine Basketball Association?

Expected Answer: Joselito Perez Marquez, better known as Joey Marquez, is a Filipino actor, comedian, politician and former professional basketball player in the Philippine Basketball Association.

Predicted Answer: Marina Augusta Pepper (née Baker; born 8 December 1967) is an English Liberal Democrat local politician, journalist, children's book author and former model and actress.

Fig. 17. Inaccurate results obtained from HDP model

Since, the results from the HDP model were not very convincing, we decided to go ahead with the remaining three models to evaluate the performance of our Question-Answering system.

VIII. LIMITATIONS

As stated in the preceeding sections, the primary objective of the project is to utilize the concepts covered during in class lectures besides learning numerous other natural language processing concepts and to build on these concepts whilst implementing the question-answering system. Hence, we explored different types of approaches to build a QA system. We would like to enlist some of the limitations of the QA system that we built using the four distinct approaches: cdQA model pre trained on the SQuAD dataset, LDA, HDP and Unsupervised-InferSent model.

- Considering the fact that the LDA model predicts based on the topics learnt from the corpus, the implemented Latent Dirichlet Allocation (LDA) model does not seem to be very accurate in predicting answers for short and less-descriptive questions, where it does not get a chance to understand the topics.
- The Unsupervised-InferSent approach correctly predicts the answers even for short and less-descriptive questions. However, even this level of predictability is below par in comparison to the cdQA model pre-trained on the SQuAD dataset.
- The HDP model did not seem to give convincing results while predicting answers to the queries, hence we would like to explore more on this model in the future.
- The system is trained on approximately 1600 instances and has less number of training features. In order to increase the overall accuracy and predictability of the question answering system, we might want to consider

more training instances and also increase the dimensionality of the training data.

IX. Conclusion

Our work primarily emphasizes on the utilization of four distinct approaches namely, the closed-domain Question Answering (cdQA) pipeline approach, the Latent Dirichlet Allocation (LDA) approach, the Hierarchical Dirichlet Process (HDP) approach and an unsupervised learning approach. The retriever and the reader were the two main essential constituents of the cdQA approach. The LDA approach was used to extract abstract topics that existed across our dataset. The resultant model constructed from this approach was used to predict the topic of the test questions besides, extracting the most relevant document corresponding to the test question. The HDP approach was incorporated so as to determine the optimum number of topics that existed across the dataset using posterior interference and thus, served as an extension of the LDA model. The unsupervised approach was inclusive of a Facebook sentence embedding named 'Infersent' [5] that provided semantic representation of the English sentences that existed in our dataset.

The models constructed based on the aforementioned approaches were trained on our very own dataset that has been prepared using the contents of 1,605 Wikipedia web pages of global politicians. The strategy devised to evaluate these constructed models required us to manually build and annotate a test dataset that consisted of 400 potential questions. The test dataset was structured so as to include the questions with their corresponding annotated, expected paragraphs where the answer to these questions existed. It was decided to incorporate accuracy derived from cosine similarity in our evaluation strategy to determine the similarity between the expected and predicted answers. It is imperative to note that the constructed HDP model has been disregarded in the computation of cosine similarity and accuracy since the results obtained from the model were not consistent and not statistically significant. Among the remaining three models, the cdQA model showed the highest accuracy of 93%, while the unsupervised-InferSent model and the LDA model showed accuracies of 77% and 62% respectively. Interestingly, the highly accurate cdQA model took approximately 2.5 hours to predict the answers in comparison to the other two less accurate LDA and unsupervised-InferSent models. In conclusion, the results obtained through our work imply that a trade-off between accuracy and time taken to predict the answers that needs to be achieved depends on the scenario and requirement under consideration.

X. Future Work

We would like to extend our work with the Question-Answering system and consider the following enhancements to our system that would improve the accuracy as well as the performance of our system.

- Design measures to enhance the prediction accuracy of the implemented Latent Dirichlet Allocation (LDA) model so that it can accurately predict answers for short, less-descriptive questions. We would like to consider the lda2vec learning approach, where we could combine the LDA model with a word2vec so that it can jointly learn word, documents and topic vectors [17].
- Explore Deep Learning techniques with Sequence Modeling to build this system.
- Explore how to increase the efficiency of predictions in Hierarchical Dirichlet Processing (HDP) model by tuning some hyperparameters like specifying the maximum number of topics it can learn and combining it with word-embeddings to give more accurate results.
- Significantly enhance the overall ability of the constructed models to exploit domain-specific knowledge either by increasing the dimensionality of the prepared training dataset or through feature engineering. This shall in turn positively impact the overall accuracy and predictability of the question answering system.

Acknowledgement

We would like to express our sincere gratitude to Professor Diana Inkpen for her valuable guidance and for providing us the necessary background on Natural Language Processing that helped us put our knowledge to practice for the implementation of this project.

References

[1] "Main Page." Wikipedia, Wikimedia Foundation, 27 Apr. 2020, www.wikipedia.org/.

[2] Farias, M. (2019, October 21). How to create your own Question-Answering system easily with python. Retrieved from https://towardsdatascience.com/how-to-create-your-own-question-answering-system-easily-with-python-2ef8abc8eb5

[3] gensim: topic modelling for humans. (2020). Retrieved 28 April 2020, from https://radimrehurek.com/gensim/models/hdpmodel.html

[4] Latent Dirichlet Allocation vs Hierarchical Dirichlet Process. Retrieved from https://datascience.stackexchange.com/questions/128/latent-dirichlet-allocation-vs-hierarchical-dirichlet-process

[5] Ahmad, R. (2019, July 26). Sentence Embeddings-Facebook's Infersent. Retrieved from https://medium.com/analytics-vidhya/sentence-embeddings-facebooks-infersent-6ac4a9fc2001

[6] Swalin, A. (2018, June 1). Building a Question-Answering System from Scratch- Part 1. Retrieved from https://towardsdatascience.com/building-a-question-answering-system-part-1-9388aadff507

[7] Malik, U. (n.d.). Python for NLP: Working with the Gensim Library (Part 1). Retrieved from https://stackabuse.com/python-for-nlp-working-with-the-gensim-library-part-1/

[8] Scikit-Learn. (n.d.). Perplexity not monotonically decreasing for batch Latent Dirichlet Allocation · Issue 6777 · scikit-learn/scikit-learn. Retrieved from https://github.com/scikit-learn/scikit-learn/issues/6777

[9] Asli, Celikyilmaz Hakkani-Tur, Dilek Tur, Gokhan. (2010). LDA based similarity modeling for question answering. NAACL HLT Workshop on Semantic Search. 1-9.

[10] S. Louvigné, J. Shi, Y. Kato, N. Rubens and M. Ueno, "A corporal and LDA analysis of abstracts of academic conference papers," Proceedings of the 2013 International Conference on Advanced Mechatronic Systems, Luoyang, 2013, pp. 412-416.

[11] Anupriya, P. Karpagavalli, S.. (2015). LDA based topic modeling of journal abstracts. 1-5. 10.1109/ICACCS.2015.7324058.

[12] Scala NLP (n.d.). Retrieved from http://www.scalanlp.org/

[13] Rule-based matching · spaCy Usage Documentation. (n.d.). Retrieved from https://spacy.io/usage/rule-based-matchingentityruler

[14] Duan, Jianyong Ai, Yamin li, Xia. (2015). LDA topic model for microblog recommendation. 185-188. 10.1109/IALP.2015.7451562.

[15] C. Troussas, A. Krouska and M. Virvou, ”Automatic Predictions Using LDA for Learning through Social Networking Services,” 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), Boston, MA, 2017, pp. 747-751.

[16] Facebookresearch. (n.d.). facebookresearch/InferSent. Retrieved from https://github.com/facebookresearch/InferSent

[17] Xu, J. (2018, December 20). Topic Modeling with LSA, PSLA, LDA lda2Vec. Retrieved from https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05