

A REPORT
ON
IMPLEMENTING A REST API FOR GIVING
RELEVANT RESULTS BASED ON THE
USER QUERY USING NATURAL
LANGUAGE PROCESSING



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

BY

AKSHAY VALSARAJ

2018B1A70608G

At

PARENTLANE

Bangalore

June, 2014

A REPORT

ON

IMPLEMENTING A REST API FOR GIVING

RELEVANT RESULTS BASED ON THE USER

QUERY USING NATURAL LANGUAGE

PROCESSING

BY

AKSHAY VALSARAJ

2018B1A70608G

Prepared for

Partial fulfillment of the course Practice School-I BITS-F221

At

PARENTLANE, Bangalore

A Practice School - 1 station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI



June, 2020

1.1 Practice School Division

Station: Parentlane

Centre: Bangalore

Duration: 6 weeks

Date of Start: May 18th, 2020

Date of Submission: June 7th, 2020

Title of Report: Implementing a REST API for giving relevant results based on the user query using natural language processing

Name of the students: Akshay Valsaraj - 2018B1A70608G

Name and Designation of Experts: Vijay Anand MV - Co-founder and CEO

Neeraj Kumar Gupta - Co-founder and CTO

Names of PS Faculty: Prof. Raja Vadhana

Project Areas: Deep Learning and network services

Key Words: Natural language processing, word vectors, word embeddings, Transformers, REST API

Abstract: Natural Language Processing, NLP for short, plays an important role in everyday applications such as text mining, chatbots and dialogue systems. NLP has also been implemented in Search engines that map the relevance of a document to a query. The goal of this project is to implement a Deep Learning Model using transformers that understands the semantics of the query given by the user and then set up a REST API so that relevant answers and articles could be provided based on the questions asked by the user and their individual parameters.

Signature of the Student

Signature of the PS Faculty

Date:

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI (HYDERABAD)**

PRACTICE SCHOOL DIVISION

Response Option Sheet

Station: Parentlane

Center: Bangalore

ID No. & Name(s): Akshay Valsaraj -

2018B1A70608G

Title of the Project: Implementing a Rest API for Giving Relevant Results Based On
the User Query Using Natural Language Processing

Usefulness of the project to the on-campus courses of study in various disciplines. Project should be scrutinized keeping in view the following response options. Write Course No. and Course Name against the option under which the project comes.

Refer Bulletin for Course No. and Course Name.

Code No.	Response Option	Course No.(s) & Name
1.	A new course can be designed out of this project.	NO
2.	The project can help modification of the course content of some of the existing Courses	NO
3.	The project can be used directly in some of the existing Compulsory Discipline Courses (CDC)/ Discipline Courses Other than Compulsory (DCOC)/ Emerging Area (EA), etc. Courses	NO
4.	The project can be used in preparatory courses like Analysis and Application Oriented Courses (AAOC)/ Engineering Science (ES)/ Technical Art (TA) and Core Courses.	NO
5.	This project cannot come under any of the above mentioned options as it relates to the professional work of the host organization.	YES

Signature of Student
Date:

Signature of Faculty
Date:

ACKNOWLEDGMENT

We would like to thank the **PS Division** for providing an interesting course like PS-I which gives us professional exposure. We are grateful to **PARENTLANE** for providing an opportunity to intern here. **Mr. Vijay Anand** and **Mr. Neeraj Kumar Gupta** have been invaluable in providing guidance and mentorship during the course of the project. We are also indebted to **Prof. Raja Vadhana** for encouraging and guiding us during this course.

TABLE OF CONTENTS

1 INTRODUCTION.....	8
2 NATURAL LANGUAGE PROCESSING.....	9
2.1 SEMANTIC TEXTUAL SIMILARITY	9
2.2 WORD EMBEDDINGS.....	10
2.3 TRANSFORMERS.....	10
2.4 THE BERT MODEL.....	11
2.5 T5 MODEL.....	12
2.6 TEXT RANK ALGORITHM.....	12
3 REST API.....	13
3.1 FLASK	13
4 ABOUT THE PROJECT.....	14
5 TASK 1: PREDICTING RELEVANT QUERIES BASED ON THE USER QUERY	15
5.1 DATA PREPROCESSING.....	15
5.1.1 NORMALIZATION.....	16
5.1.2 LEMMATIZATION.....	17
5.2 WORD EMBEDDING WITH THE ROBERTA MODEL	17
5.3 COSINE SIMILARITY	17
5.4 MODEL IMPLEMENTATION.....	18
5.5 QUESTION AND ANSWER SCORING.....	18

5.6 REST API IMPLEMENTATION ALONG WITH ANSWERS.....	19-20
6 TASK 2 : PREDICTING RELEVANT ARTICLES BASED ON THE USER QUERY	21
6.1 DATA PRE-PROCESSING.....	22
6.2 RoBERTa MODEL.....	22
6.3 IMPLEMENTING THE TEXT RANK ALGORITHM.....	23
6.4 ABSTRACTIVE TEXT SUMMARIZATION USING THE GOOGLE T5 MODEL.....	23
6.5 MODEL IMPLEMENTATION.....	24
6.6 REST API IMPLEMENTATION.....	25
7 CONCLUSION.....	26
8 REFERENCES	26-27

1 INTRODUCTION

In the past few years Artificial Intelligence, or AI, has become more and more pervasive in our everyday environments. From Google Search to Siri, from Spam filtering to home automation – AI is everywhere. With this rising ubiquity comes the increasing need for regular human interaction. This interaction needs to be seamless, uninterrupted and human-like. Every word or sentence that we express contain a lot of data and information that can be used to analyze our state of mind and even understand the complexities of human behavior. A person may generate millions of words and it becomes difficult for the computer to interpret the behavior of the user. This huge gap in information processing has been solved by deep learning techniques such as NLP (Natural language Processing) which gives the ability to machines to understand and read human language and derive a meaning from it .NLP has been used in a variety of fields ranging from prediction of diseases, sentiment analysis which can predict the user profile for customer targeting by advertisers. The process of understanding the text and giving the relevant searches can be very tricky as the model should be able to understand the semantics of the text and various models have been proposed with the latest advancements being done in Deep learning using Transformers.

A transformer creates word embeddings of the sentence and finds the matching texts to the query using the metric of cosine similarity .The model after development should also be available to the users so that they can interact with it .This can be achieved using web services such as REST (Representational State Transfer) API. RESTful Web services help different computers to interact such as send a request and receive a response .The outlook of the API is dictated by REST which provide a set of rules for developers to make the application on the server such that the client can easily communicate with it.

2 NATURAL LANGUAGE PROCESSING:

Human language is highly complicated, convoluted and intricate in nature. Its vagaries and subtleties are virtually endless. This makes it exceptionally difficult for programmers to make a program that can understand the semantics of the text. So much so that the Turing Test (a test developed by Alan Turing to measure a machine's ability to display intelligent, human-like behaviour) is basically a measure of language proficiency. However, recent advancements in deep learning and the rapid increase in computer processing power have allowed the field of NLP to break new ground in speech recognition, text generation, summarization, question-answering, etc. This has helped in tracking disease spread using social media, sentiment analysis and even chatbots.

2.1 SEMANTIC TEXTUAL SIMILARITY

Semantic Textual Similarity, or STS, is a subfield of NLP. It is the comparison of two pieces of text to determine their similarity in intent or meaning. This is as opposed to lexicographical similarity which is a measure of overlap in terms of word sets and word sequences. The applications of STS are wide far-reaching. They include machine translation like Google Translate, summarization, generation, short answer grading, dialog and conversational systems. The list goes on. One of the techniques used in measuring the STS of two texts is to create their word embeddings and use cosine similarity measure (which is somewhat akin to the inverse of the Euclidean distance between vectors – closer the vectors, greater the similarity).

2.2 WORD EMBEDDING:

Word Embedding is a language modelling technique in which individual words or phrases are mapped to vectors of real numbers. It is an easy and effective way of generating numerical representations of word meanings and their relationships to one another. One of the methods to generate word vectors is to use neural networks. They initially begin with a random representation of the vocabulary and over several iterations over the network, using the context (the surrounding words) of the current word, modify its vector representation. In this way, words that are similar in meaning, have similar vector representations. Word embeddings, when used as the underlying input representation, have been shown to improve performance of downstream NLP tasks such as sentiment analysis and semantic textual similarity. One of the drawbacks of Word Embedding is that multiple meanings of the same word have to be encapsulated by the same vector. Several Neural Net Architectures can be used to give rise to Word Embeddings, however, one of the models that has proven especially effective at NLP tasks is a Transformer.

2.3 TRANSFORMERS:

A Transformer is a neural network composed of 2 LSTM networks (Long Short Term Memory networks) one which acts as a encoder and another that acts as a decoder, and an attention mechanism that keeps track of the most semantically valuable words in the input sentence.

LSTM: An LSTM is made up of an RNN (Recurrent Neural Network). Unlike regular feedforward neural networks, where connections between neurons do not form a cycle, an LSTM has feedback connections as well. It is exceptionally well suited to handle a sequence of data points like a series of words (sentences) or images (video), and thus, it finds use in a transformer.

Encoder/Decoder: An encoder is a neural network that takes the input, and outputs a feature vector. These feature vectors represent the input (like word embeddings). The decoder is again a neural network (usually the same network structure as the encoder but inverted in orientation) that takes the feature vector from the encoder, and gives the closest match to the actual input or intended output.

The Attention Model: The attention mechanism was proposed as a solution to the limitations of traditional encoder-decoder structures. By keeping track of the most important parts of a sentence, it enables the transformer to handle large sentences, especially those sentences that are longer than those in the training corpus. [1]

Use: Its context sensitive encoding technique is effectively leveraged for machine translation and other NLP tasks like STS and question answering.

2.4 The BERT Model:

BERT stands for **B**idirectional **E**ncoder **R**epresentation from **T**ransformers. It was released in 2018 (paper in 2019) and improved on several state-of-the-art results in NLP. It uses two-way training of the transformer in contrast to previous efforts which either used left-to-right training or combined left-to-right and right-to-left training. [2] Hence the name bidirectional. To achieve this, Masked Language Modelling is used which hides a percentage of the occurrences of a word and asks the model to predict the word which will occur there. Google has released several variations of the model, with varying degrees of size. Due to the large amount of data required, the model has been pre-trained on the vast textual resources available on the net.[3]

2.5 T5 MODEL

The T5 model which stands for Text-To-Text Transfer Transformer model is a state of model introduced by Google which claims to explore the limits of transfer learning. It was trained on the Colossal Clean Crawled Corpus (C4) and achieves higher accuracy than many other NLP models and is widely used for question-answering and text summarization. For our current project we plan to use the T5 model for abstract text summarization to shorten long text documents into sentences that are only about 30 to 100 words long .This would be more useful for content analysis as creating an embedding for each article which is about 1000 words is very memory intensive.[5]

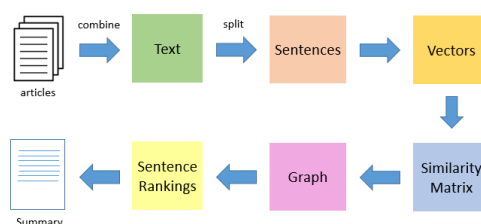
2.6 TEXT RANK ALGORITHM

One major drawback of the T5 model is that it can't accept more than 512 tokens so we implemented the Text rank algorithm on each article to get the most relevant sentences such that the total word limit of all sentences in the article is never more than 512 words. The text rank algorithm finds how similar each sentence is compared to other sentences in each of the articles.

Definition 1. Given S_i, S_j two sentences represented by a set of n words that in S_i are represented as $S_i = w_1^i, w_2^i, \dots, w_n^i$. The similarity function for S_i, S_j can be defined as:

$$Sim(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \& w_k \in S_j\}|}{\log(|S_i|) + \log(|S_j|)}$$

Each sentence is considered as a node and each sentence is connected to other sentences using a edge or a vertex. The weight on the edges is found by using the similarity function. Text rank is based on the PageRank algorithm used by google to display web results. [6]



The sentences are then sorted based on the scores and the first n sentences within m words are taken to be part of the text summary

3 REST API

Rest stands for **R**epresentational **S**tate **T**ransfer, It defines a set of rules while creating web services and the API is an application programming interface which provides the client-server interface. The various methods to perform different operations in RESTful APIs are

- GET – To retrieve the resource
- POST – To create new resources
- PUT – To update the existing resource
- DELETE – To delete resources

The data is sent through a specific URL which acts as a request. Its components are – endpoint, method, headers and the body.

We have implemented the JSON format for sending and requesting data for the REST API.

3.1 FLASK

Flask is a web framework based on Python which provides the basic tools and libraries for easier development of web applications. It is a micro-framework and doesn't have extensive external library dependencies. It is well suited for small scale projects. Many applications such as Pinterest and LinkedIn are built on the FLASK framework.

Flask makes use of the Jinja2 template engine which helps in providing programming constructs in HTML. It also uses Werkzeug which is a utility library for python and also provides Web Server Gateway Interface applications (WSGI).

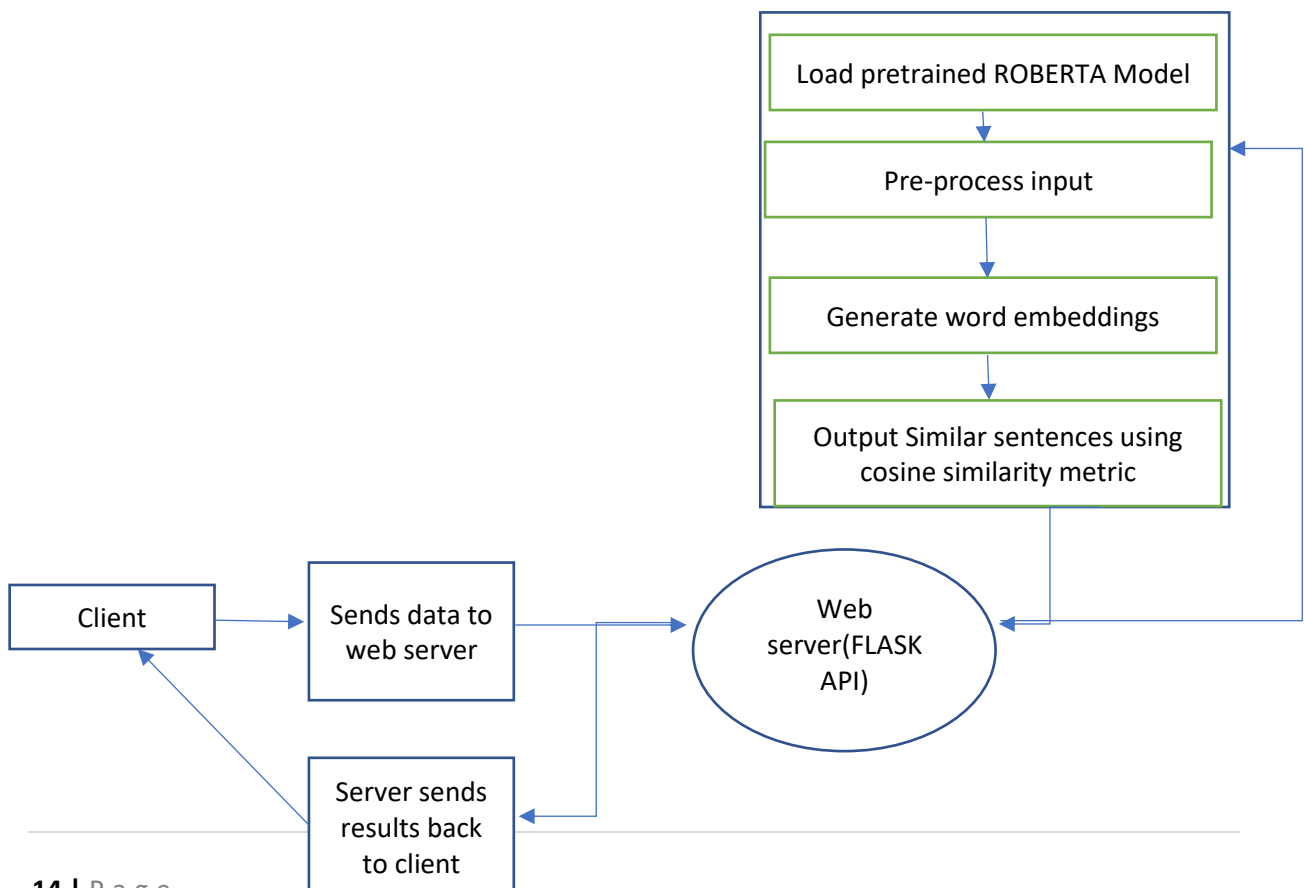
4 ABOUT THE PROJECT

The project can be divided into two parts:-

- The first task was to return similar queries asked by previous users so that the user could refer to their responses and thereby prevent duplicate queries. Moreover the responses are also classified based on the personal parameters of the user such as age and pregnancy.
- The second task was to give relevant articles written by experts based on the user query so that the user could get an expert opinion.

Both the models were implemented on a REST API and The SQL data base of both the tasks was given by the company. The model was implemented using a version of BERT and the API framework was built using flask.

The user sends a query as a request. It is pre-processed and its word embedding is compared with the word embeddings of all the questions from the database. Similar queries, found using cosine similarity are sent as response to the client.



5 TASK 1 : PREDICTING RELEVANT QUERIES BASED ON THE USER QUERY

Our current work focuses on semantic textual similarity which helps to compare texts to determine their similarity. We have implemented a version of the RoBERTa (itself an improvement on BERT) model as it was shown to have better performance than other models on the GLUE benchmark (The General Language Understanding Evaluation benchmark is a collection of resources for the evaluation of several critical NLP tasks). The questions in the dataset were pre-processed and comparison was made using cosine similarity.

5.1 DATA PREPROCESSING

The raw data given in the database was rife with human errors and need to be processed into normalized text (a process called lexical normalization). It also helps in filtering out string/text segments that do not contribute to the accuracy and performance of the model.

The database provided had 50060 rows and 35 columns. The rows represent the various queries asked by the users and the columns contained additional relevant user information and data about the query like time of posting, age range of child, likes, number of answers, etc. After extensive evaluation it was decided that only 5 columns – namely user query, user child age, minimum age group, maximum age group and pregnancy type gave useful information and the rest of the columns were dropped. A pregnancy type of 0 indicates that the user is not pregnant and a pregnancy type of 1 indicated the user is pregnant. The age group column helps in giving relevant user queries that are within the group since treatments are age sensitive, especially for young children. For pregnant children, each trimester is a distinct age group, while up to 1 year old the groups are 3 months long. After the first year each category is the length of a year. All the queries which were in languages other than English were removed as the model was

pretrained only in English. The database now contains only 43081 rows and 5 columns.

INDEX	USER QUERY	MINIMUM AGE GROUP (days)	MAXIMUM AGE GROUP (days)	CHILD AGE	PREGNANCY TYPE
1	Can I give cerlac to baby ??????	90	179	94	0
2	My baby has cold	180	269	229	0
3	my baby is not sleeping	0	89	60	0
4	should we bath babies daily	90	179	161	0
5	My baby boy has running fever	90	179	149	0

Table 1: representation of the first five rows of the database after dropping the unnecessary columns and the queries which are not in English

The various techniques employed for processing both – the database queries and those newly entered by the user – are:

5.1.1 NORMALIZATION

One of the important features of data cleaning is to remove noise so that the model can detect the patterns easily and give relevant queries. The user query contains a lot of noise in the form of special characters, punctuation and http tags such as '
', ' sp', '
', etc. Link related strings like 'http', 'www', 'YouTube' etc. have also been removed as they do not contribute to the meaning of the sentence.

Stop word removal, though often a part of text normalization, was excluded because of the importance of stop words (like 'no', 'to', 'for', etc.) in interpreting and establishing semantic relationships between words. Their removal would make the sentence essential meaningless to the model.

5.1.2 LEMMATIZATION

Lemmatization helps in reducing words to their root form. For examples it resolves words like 'is' and 'are' to its root form 'be'. We have used the NLTK library of python which is based on the WordNet database. We have also used POS tagging which categorizes words into their respective parts-of-speech like nouns, verbs, adjectives, etc. This information helps make the lemmatization context dependent and increases its effectiveness.

5.2 WORD EMBEDDING WITH THE ROBERTA MODEL

Even though BERT produces good results, it can be improved further by optimizing it and using large batch training sizes. RoBOERTa (A Robustly Optimized BERT Pretraining Approach from Facebook) uses 160GB for pre training which includes the pre trained text from BERT, hence it outperforms BERT and other models according to the GLUE benchmark results. RoBERTa doesn't use the NSP method of BERT but introduces dynamic masking which changes the sentence token during every training epoch which prevents over fitting. RoBOERTa follows the BERT architecture and uses 24 input layers, 1024 hidden layers and 16 attention head with a total of 355 million parameters.[4]

5.3 COSINE SIMILARITY

After generating the word embedding as vectors for each sentence it is necessary to have a metric to measure the similarity of the sentences. This can be achieved by using cosine similarity which considers the angle between the sentence vectors. A cosine angle of 0 would determine the sentences are identical while a cosine angle of 90 degrees would determine that they are completely different sentences.

5.4 MODEL IMPLEMENTATION

After accepting the sentence it is subjected to the same pre-processing and a word embedding is created using the stored model. Then, the cosine similarity between this and every question in the database is calculated. 5 questions from the database that have the highest cosine similarity are then displayed only if they are greater than a pre-defined threshold. If the cosine similarity is lower than the threshold then the appropriate result is displayed.

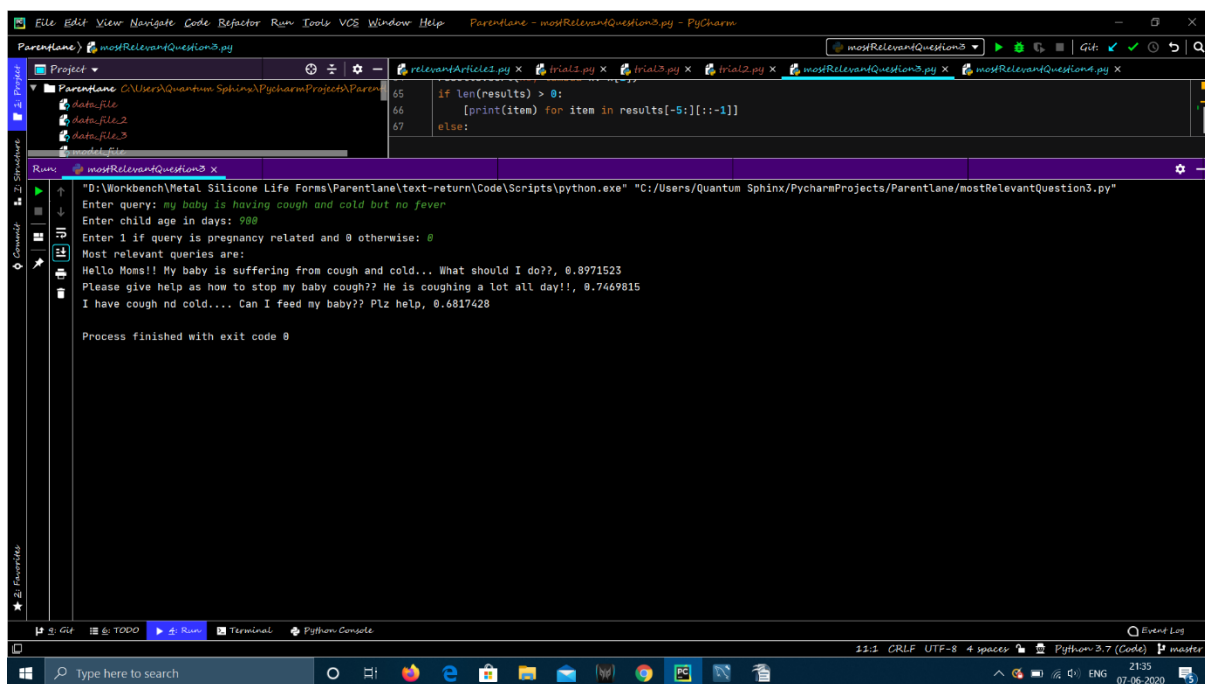


Figure 2 – The model without the REST API. It accepts the query, the age and pregnancy type and outputs the results which cross the set similarity threshold.

5.5 QUESTION AND ANSWER SCORING

After predicting the most relevant queries ,our next plan of action was to even output the most relevant answer to each question. The answers database was provided by the company for this purpose. The important columns from the answers database that were important for our task was the content of the answer, question id and created by. Each answer also has a question id which gives us information about the question it belongs to. Moreover the ‘created by’ is also an important field as many answers are given by experts and such answers have a higher

weightage than the other results.

Index	answer	question_id	created_by
1	These are the tips to increase ur babies weight	bfd926e6-be79-4552-a768-988c213f990e	e29ca5e2-4be5-4d43-be19-12d929a25ff6
2	Yes babies lose their weight in initial days a...	f33f57e6-5d8d-4240-9b58-216a947639e8	9c4d8833-7d10-4985-97fa-7b7798c172b4
3	Hi dear, there is no need of honey at this age...	ad907004-24b5-433a-8776-b94d98797a10	03923462-1887-4c19-9e90-c9cfe86d6797
4	Your baby is about to enter his eight month. W...	65870f13-3a97-49b3-b2de-56665c50f740	f9762cab-a407-4523-885c-f82d0f6ebed6
5	If ur baby is active and achieving all the mil...	1b10f2f1-ac21-45ac-b167-a8ab9955873e	ed804a8e-3fc1-414f-bf62-f6aabcf31d79

Table 2: representation of the first five rows of the answers database after dropping the unnecessary columns and rows

To assign a score for each answer we used an algorithm provided by the company which gave the score for each answer based the number of words each answer has and whether the answer was given by an expert. An expert is identified by a unique id in the ‘created by’ column.

The question were also assigned a score based on the number of answers that are there for each question and if it has an expert answer id ,it is given a very high weightage and assigned a score of 100. The questions which have no answer are given a score of 0.

5.6 REST API IMPLEMENTATION ALONG WITH ANSWERS

The model was then finally implemented along with the REST API such that when the user types the query the top 5 relevant queries asked by previous users were shown along with the answer with the highest score for that question.

2 endpoints namely ‘/’ and ‘/predict’ was created .The endpoint ‘/’ connected to the home.html which contained a text box for the user to type the query and another text box to

type the age of the baby in days or the number of days the mother has been pregnant. An option of a checkbox was added to classify the results if the user is pregnant.

The REST API and the model was tested by providing 10 unique questions from the Parentlane app and it gave satisfactory results on all occasions and the REST API was also finally hosted on the amazon web server provided by the company and tested by the company instructors .

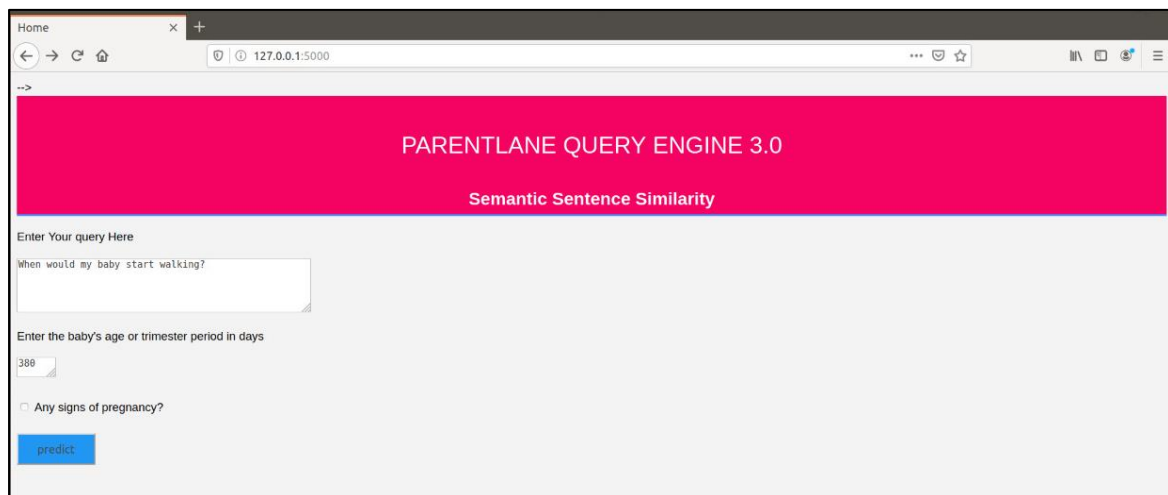


Figure 2 – Home page for user to type the query and other personal parameters such as age and pregnancy for better results

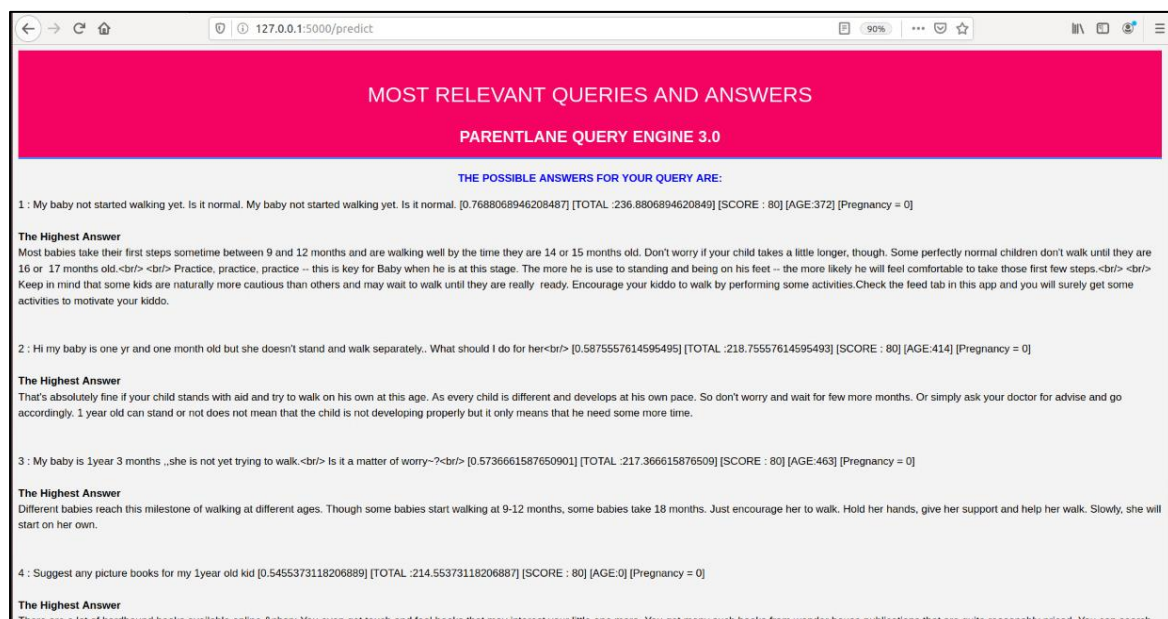


Figure 3 – Result page to show the top 5 relevant queries along with their top answer

6 TASK 2 : PREDICTING RELEVANT ARTICLES BASED ON THE USER QUERY

Our second task also focussed on semantic textual similarity but on a large scale as now we have to find how similar each query is compared each article. The content database was provided by the company consisted of 11823 articles and had 45 columns. The important field for this project is only the column which has the content, title and the URL of the article. Most of the articles have an average of 500 words.

Index	Title of the article	Article content	URL
1	Your ideal diet at this stage	<p>At week 10 of your pregnancy, this is what ...	NaN
2	Aloo Paratha	<p>Aloo paratha is a famous Punjabi food item ...	https://www.parentlane.com/recipes/child-food-...
3	Start counting your baby's movements now	<p><span style="font-family: Arial; font-size:...	NaN
4	Teach Your Kid To Adapt To Sudden Changes In S...	<p><span style="font-family: Arial; font-size:...	https://www.parentlane.com/parenting/child-dev..
5	Basic Yoga Poses to relieve pregnancy cramps	<p>Category: yoga </p><p>Benefits...	NaN

Table 3: representation of the first five rows of the contents database after dropping the unnecessary columns

6.1 DATA PREPROCESSING

The content data provided was taken directly from the webpage ,hence it had a lot of HTML tags which were all removed along with punctuation. The text wasn't lemmatized as we are using the text for text summarization and lemmatization would not give good results while applying the text summarizer model. Any article which was less than 100 words was removed as it wasn't an article written by an expert and not verified. Some of the rows were empty for the article content as the information was given in the form of an image, hence all these rows were removed and the database now contains only 10335 rows and 4 columns.

6.2 RoBERTa MODEL

We implemented the RoBERTa model that we had used on the query engine to see the results .The model was implemented on the content data converting each word of each article into a token. The RoBERTa model acted as a baseline model so that we could use other models and improve upon it.

```
Query: when will my baby walk

Top 10 most similar sentences in corpus:
Know Your Baby At Week 1 (Cosine Score: 0.6388)
When Will My Baby Start Walking? (Cosine Score: 0.6100)
Your Child's New Developments at 1 month! (Cosine Score: 0.6096)
Glance of Your Child Development from 1 Month to 1 Year (Cosine Score: 0.5737)
Know Your Baby At Week 5 (Cosine Score: 0.5681)
Milestones of your baby (Cosine Score: 0.5647)
Your Child's Speech Milestones (Cosine Score: 0.5616)
Your Child's New Developments at 6 Months! (Cosine Score: 0.5614)
Your Child's New Developments at 5 Months! (Cosine Score: 0.5570)
How Long should your Baby be Awake? (Cosine Score: 0.5562)
```

Figure 4 – Model implementation of the content analysis using the previous RoBERTa model

The results of the RoBERTa model wasn't that accurate as it didn't give the expected answers and gave varying results. We realised that the accuracy of the model decreased as the length of the corpus increased hence our next plan of action was to reduce unnecessary information in the article and feed the data into the model in a more concise format for better results.

6.3 IMPLEMENTING THE TEXT RANK ALGORITHM

The text rank algorithm was then implemented on all the 10335 articles such that if a text had more than 512 words the text rank reduced the sentences of the article such that it was below the limit set as the pre trained weights of T5 text summarizer model couldn't handle articles which are more than 512 words. The text rank algorithm was implemented using the gensim library of python and the text ranked articles were then stored in a new column

6.4 ABSTRACTIVE TEXT SUMMARIZATION USING THE GOOGLE T5 MODEL

The pretrained model was obtained from hugging face and it had 60 million parameters with 6 layers, 512 hidden states, 2048 feed-forward hidden-state and 12 attention heads which was trained on the Colossal Clean Crawled Corpus (C4). The text summarization was done on the pre trained model 't5-samll' as text summarization is very GPU intensive task and it took 3 hours for summarizing all the articles which were text ranked and it was stored in a new column.

6.5 MODEL IMPLEMENTATION

The word embeddings were then created from the summarized text using the RoBOERTa (A Robustly Optimized BERT Pretraining Approach from Facebook) model and cosine similarity is used as a metric.

After accepting the query from the user it is subjected to the same pre-processing and a word embedding of the query is created and cosine similarity between this and every summarized text in the database is calculated and the title of the top 10 articles having the highest score are displayed.

```
[24] import scipy
number_top_matches = 10
for query, query_embedding in zip(queries, query_embeddings):
    distances = scipy.spatial.distance.cdist([query_embedding], sentence_embeddings, "cosine")[0]
    results = zip(range(len(distances)), distances)
    results = sorted(results, key=lambda x: x[1])

    print("\n\n=====\n\n")
    print("Query:", question)
    print("\nTop 10 most similar sentences in corpus:")

    for idx, distance in results[0:number_top_matches]:
        print(ranked_content['title'][idx], "(Cosine Score: %.4f)" % (1-distance))
```

=====

Query: home remedies for cough and cold for my baby

Top 10 most similar sentences in corpus:

- Watch Video To Know Home Remedies To Cure Your Baby's Cough & Cold (Cosine Score: 0.7528)
- Throat Infection in Babies: Causes, Symptoms, Remedies (Cosine Score: 0.7306)
- How to Take Care of Your Baby in the Winters? (Cosine Score: 0.7142)
- Baby coughing At Night Time: Causes and Remedies (Cosine Score: 0.7027)
- When Is It Okay To Use Home Remedies On Babies? (Cosine Score: 0.6976)
- Grandma's handy tips: Home remedies for cough & cold, fever, and more (Cosine Score: 0.6582)
- Top 5 Home Remedies for Chest Congestion in Babies and Infants (Cosine Score: 0.6550)
- Do Not Opt For Strong Medicines At The First Sign Of Your Kid's Discomfort (Cosine Score: 0.6518)
- Bringing your baby home for the first time: What to expect (Cosine Score: 0.6391)
- Child Falling Sick Frequently? Here's What You Can Do (Cosine Score: 0.6359)

Figure 5 – Model implementation of the content analysis using text rank and text summarization using the T5 model along with word embedding using the RoBERTa model

6.6 REST API IMPLEMENTATION

The model was then finally implemented along with the REST API such that when the user types the query the top 10 relevant articles are displayed along with the URL and a short summary of the article similar to a google search.

2 endpoints namely '/' and '/predict' was created .The endpoint '/' connected to the home.html which contained a text box for the user to type the query

The REST API and the model was tested by providing 10 unique questions from the Parentlane app and it gave satisfactory results on all occasions and the REST API was also finally hosted on the amazon web server provided by the company and tested by the company instructors.

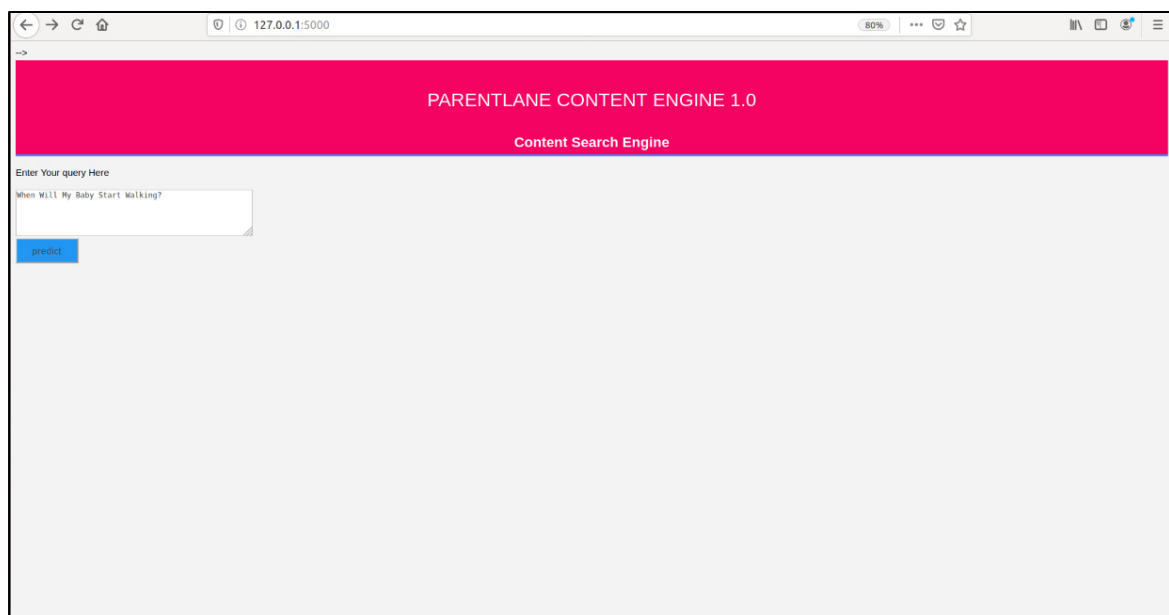


Figure 6 – Home page for user to type the query

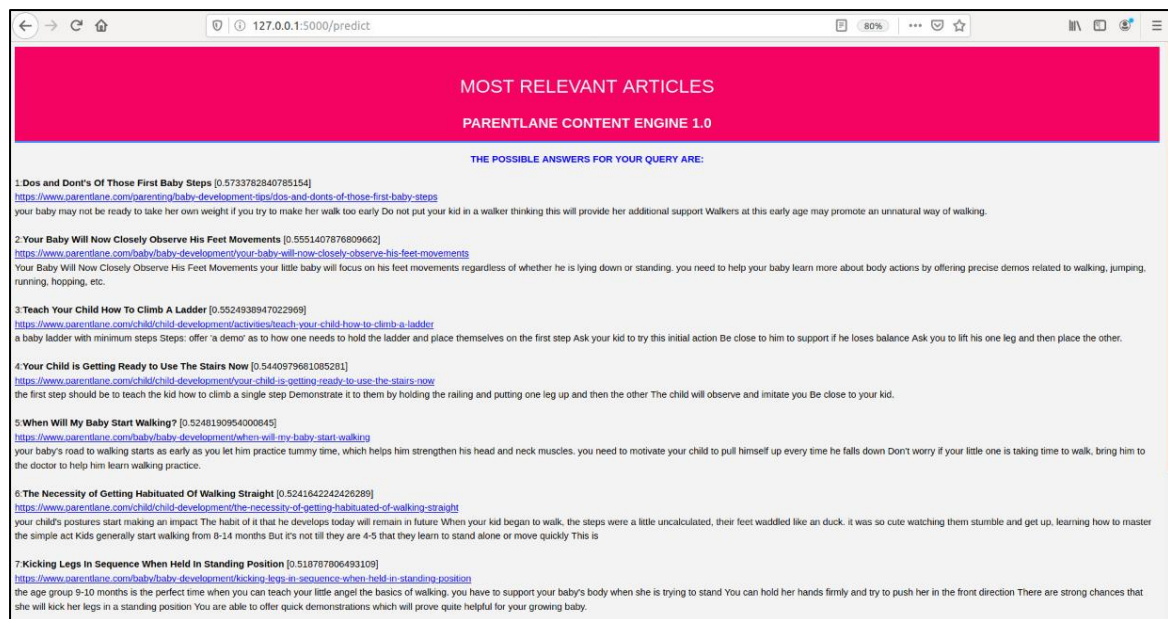


Figure 7 - Result page to show the top 10 relevant articles along with their URL and a short snippet of the contents of the article similar to a Google Search.

7 CONCLUSIONS:

The current state of STS is such that it misses some and gets some of the subtleties and nuances of human speech, though far outpacing traditional lexical approaches. Accounting for the added hurdle of the text dataset not being standardized (like Wikipedia), our results mirror the above conclusion.

8 REFERENCES

- [1] Vasvani et. al. *Attention Is All You Need*. ArXiv preprint arXiv:1706.03762, 2017
- [2] Devlin et. al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. ArXiv preprint arXiv:1810.04805, 2019
- [3] Reimers et. al. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. ArXiv preprint arXiv:1908.10084, 2019
- [4] Liu et. al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. ArXiv:

1907.11692, 2019

[5] Liu, Colin Raffel Noam Shazeer Adam Roberts Katherine Lee Sharan Narang Michael Zhou Wei Li Peter J. "*Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.*" *arXiv* (2019): 53.

[6] Erkan & Radev Mihalcea, R., & Tarau, P. (2004). Textrank: *Bringing order into texts*. In Lin, D., & Wu, D.(Eds.), *Proceedings of EMNLP 2004* , pp. 404{411 Barcelona, Spain. Association for Computational Linguistics.