QUESTION ANSWERING SYSTEM USING NATURAL LANGUAGE PROCESSING TECHNIQUES

Sai Manasa Vedantam
(SXV200055)

Janam Parikh
(JSP180001)

PROBLEM DESCRIPTION

On a day-to-day basis, the use of interactive, intelligent assistants which can respond to human statements is increasing. Our project focuses on building a Question-Answering System relying on the Natural Language Processing Techniques. We aim to answer WHAT, WHEN and WHO type questions. Our system uses 30 articles from the Stanford Question Answering Dataset (SQuAD) and as the question domain can't be beyond this, ours is a Closed-Domain Question-Answering System. The details regarding our approach, NLP features, techniques used and problem domain are discussed in the coming sections.

PROPOSED SOLUTION

To implement our Question Answering System (QA-System), we made a literature study and were inspired by a few ideas from the IEEE paper QA-System. However, we used that as a kickstart and took innovative steps as and when needed. Briefly, our solution reads articles one-by-one, tokenize into sentences and then to words, extract features required and store them. Then, we used the features obtained by passing through our NLP pipeline and used, inverted indexing each sentence using the Elastic Search. Later, we performed a search on the indexed document using a much deeper pipeline with varied weights associated with each feature and obtained top 10 ranked results from which we used the top solution as the final answer to the question posed. For this, we divided the solution into three tasks.

- TASK 1

In this task, we focused on passing the articles through our NLP pipeline and extract several features per each sentence in each article. However, we used only some **useful** features from the ones that we extracted. We decided upon **useful** features based on our analysis on which combination of features generated what type of results, accuracy on the validation set, level of complexity that can be handled etc. Our NLP pipeline takes articles as input in training and generates respective features in training while in testing, we pass the question through this pipeline for feature extraction.

In the following, we have the list of features extracted and the importance associated with each of them in detail:

1. **Word and Sentence Tokens :** They help to understand the context & develop NLP model
2. **Lemmas :** They help us to group different forms of word so that any version of base word can be used in the query and retrieval.
3. **Part-of-Speech Tags (POS) :** They help us to build Parse trees and can be used as a linguistic criteria.
4. **WordNet Features :** We extracted the following features from WordNet.
    a. **Synonyms :** Helps in handling exact substitutes for a word with the same meaning.
    b. **Hypernyms :** Helps in extracting the included meaning of the words.
    c. **Hyponyms :** Helps us to find the subcategory of a generic word.
    d. **Meronyms :** Helps us to identify the constituent part of something.
    e. **Holonyms :** Helps us to know the whole thing or group to which something belongs.
5. **Dependency Parsing :** This step helps us to obtain the relationship between different words of a sentence.
6. **Head Generation :** This helps us to understand the essence of the sentence.
7. **Stems :** They help us to reduce the inflectional forms of words by considering the roots.
8. **Named Entities :** They help us to identify the key elements in the sentence like people, place, time, year etc.

After extracting the features, they were stored in the dictionary data structure. This was done so that these features can be indexed efficiently in ElasticSearch.

- TASK 2

This task deals with indexing the output obtained from the NLP pipeline using ElasticSearch and using that to build better query criteria to obtain the most sensible output for the question posed to the system. To decide the query criteria, we use accuracy obtained on the validation dataset as the metric and try adding and removing features, increasing and decreasing importance associated with each feature. We built a deeper NLP pipeline which helps us to filter out irrelevant sentences quite early.

- TASK 3

This stage involves generating answers for a list of questions with different levels of complexity passed as an input to the pipeline and generating the most sensible answer obtained from our model. This completes the end-to-end implementation of our Closed-Domain Question-Answering System.

IMPLEMENTATION

In this section, we discuss the tools, technologies, frameworks etc. we used to implement our solution and explain in detail why each of them is used.

Tools & Technologies:

Different parts of our code require different tools for several purposes ranging from basic tokenization to indexing, searching and obtaining the query's answer in the required format.

Programming Language :

- Python : As python has a lot of freely available packages for Natural Language Processing which have the capacity to lessen the burden of implementing more obvious functions. This helped us to focus on what is needed for the project and work on improvising the model.

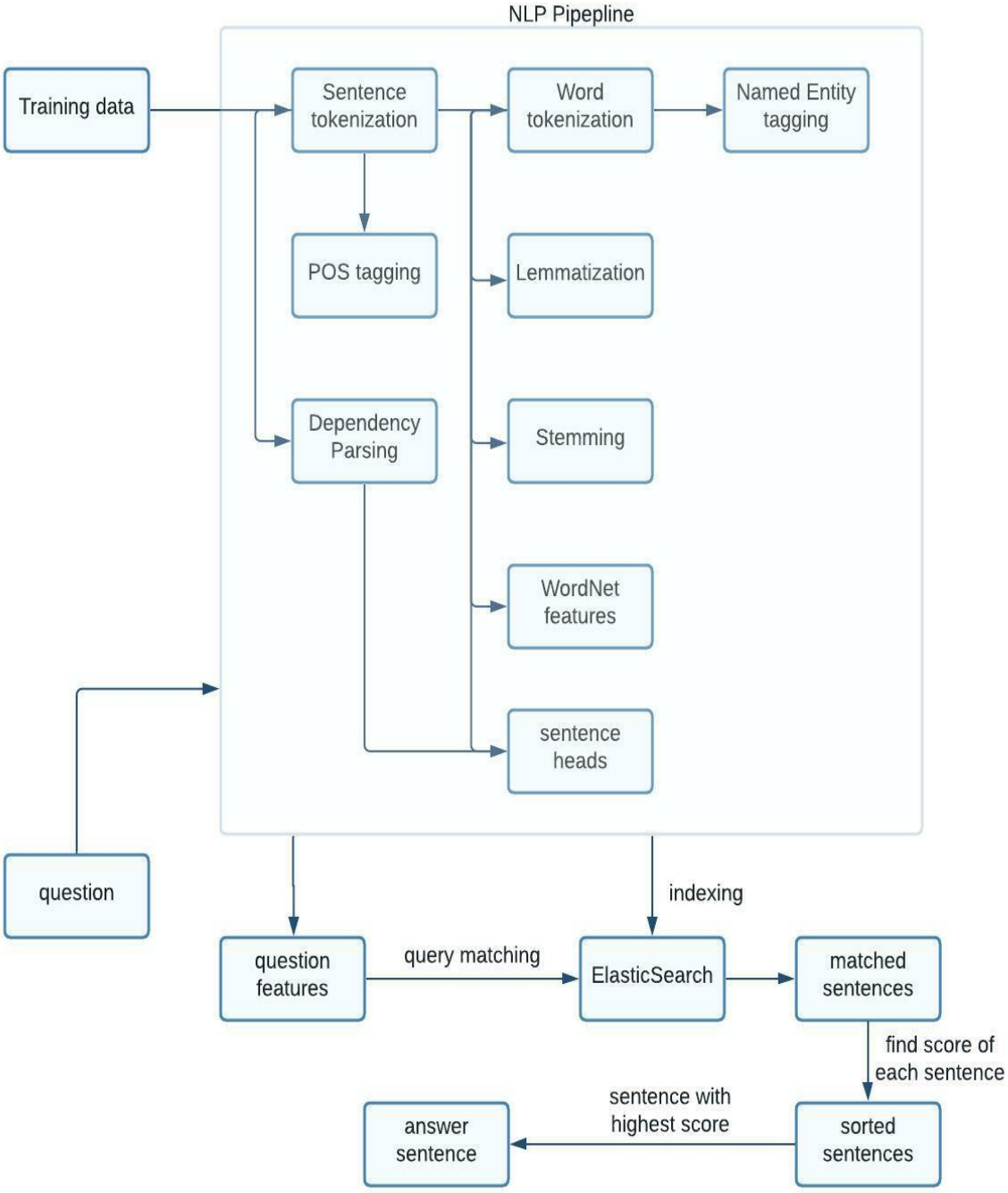
Tools Used:

- Jupyter Notebook
- Visual Studio Code
- ElasticSearch
- Lucidchart
- Github

Libraries Used:

- NLTK (sentence tokenization, word tokenization, lemmatization, stemming, POS tagging, WordNet features)
- Stanford's CoreNLP Parser (dependency parsing, Named entity tagging)
- Elasticsearch module for fetching requests from the server
- Python's 'csv' library for output formatting
- Python's WordNet library
- Lesk for obtaining the best sense based on the context
- PorterStemmer for stemming
- WordNetLemmatizer for extracting lemmas from the sentence
- Python's dateparser for parsing times in the when case
- Python's ast to process the validation dataset and obtain questions from it

MODEL ARCHITECTURE



RESULTS AND ERROR ANALYSIS

In this section, we first describe the issues we faced while completing the project, the way we addressed them and the alternatives we explored during inevitable situations. Later, we brief on the analysis we made while using features in isolation and together and the way we reached the best possible features.

- ISSUES FACED

1. Problem with CoreNLP Parser setup:

- Janam faced an issue with the setup of CoreNLP on his Windows computer. He tried a couple of solutions ranging from updating the config files to changing the maximum stack allocation for running the CoreNLP server.
- In the very extreme scenario, we tried to shift to spacy instead of CoreNLP parser.
- Finally, he was able to resolve this issue by removing the 32-bit version of Java on his laptop and installing the 64-bit version and made CoreNLP running well.

2. Problem with ElasticSearch:

- Manasa faced trouble with getting ElasticSearch work for indexing.
- There are no issues with system compatibility, version and other dependencies.
- The only trouble is that whenever elasticsearch is set up from the code, it raises SSL errors. It is due to the lack of a verification certificate.
- To fix this, new context is used to make certification verification optional.

- RESULT ANALYSIS

Answers obtained for various questions before and after using specific features, accuracy on validation data, literature reading done to obtain a better and reliable model are presented on the next page.

Features	Accuracy	Question Type / Dataset
"lemma^2.2", "ner_tag^1.8", "synonyms^1.9", "holonyms^0.2", "meronyms^0.2", "hypernyms^0.4", "hyponyms^0.4", "head_word^1.6", "tags^0.2"	60% 80% 80% 73.33% 50.26%	WHAT (10) WHO (10) WHEN (10) ALL (30) Validation data (2505)
lemma^1.2", "ner_tag^2.5", "synonyms^1.9", "holonyms^0.2", "meronyms^0.2", "hypernyms^0.4", "hyponyms^0.4", "head_word^0.6", "tags^0.2	60% 70% 70% 66.67% 46.75%	WHAT (10) WHO (10) WHEN (10) ALL (30) Validation data (2505)
lemma^0.1", "ner_tag^1.0", "synonyms^2.0", "holonyms^2.0", "meronyms^2.0", "hypernyms^2.0", "hyponyms^2.0", "head_word^0.6", "tags^2.0	70% 70% 80% 73.33% 47.07%	WHAT (10) WHO (10) WHEN (10) ALL (30) Validation data (2505)

lemma^5.0", "ner_tag^5.0", "synonyms^2.0", "holonyms^2.0", "meronyms^2.0", "hypernyms^2.0", "hyponyms^2.0", "head_word^5.0", "tags^5.0	60% 80% 70% 70% 49.42%	WHAT (10) WHO (10) WHEN (10) ALL (30) Validation data (2505)
lemma^10.0", "ner_tag^0.1", "synonyms^0.1", "holonyms^0.1", "meronyms^0.1", "hypernyms^0.1", "hyponyms^0.1", "head_word^0.1", "tags^0.1	60% 70% 70% 70% 49.02%	WHAT (10) WHO (10) WHEN (10) ALL (30) Validation data (2505)
lemma^0.1", "ner_tag^10.0", "synonyms^0.1", "holonyms^0.1", "meronyms^0.1", "hypernyms^0.1", "hyponyms^0.1", "head_word^0.1", "tags^0.1	60% 80% 50% 60% 43.39%	WHAT (10) WHO (10) WHEN (10) ALL (30) Validation data (2505)

lemma^0.1", "ner_tag^0.1", "synonyms^10.0", "holonyms^10.0", "meronyms^10.0", "hypernyms^10.0", "hyponyms^10.0", "head_word^0.1", "tags^0.1	70% 70% 70% 70% 44.51%	WHAT (10) WHO (10) WHEN (10) ALL (30) Validation data (2505)
lemma^0.1", "ner_tag^0.1", "synonyms^0.1", "holonyms^0.1", "meronyms^0.1", "hypernyms^0.1", "hyponyms^0.1", "head_word^10.0", "tags^0.1	20% 30% 0% 16.67% 5.99%	WHAT (10) WHO (10) WHEN (10) ALL (30) Validation data (2505)
lemma^0.1", "ner_tag^0.1", "synonyms^0.1", "holonyms^0.1", "meronyms^0.1", "hypernyms^0.1", "hyponyms^0.1", "head_word^0.1", "tags^10.0	60% 80% 60% 66.67% 41.76%	WHAT (10) WHO (10) WHEN (10) ALL (30) Validation data (2505)

OBSERVATIONS ON VARIOUS QUESTIONS

In this section, we listed a set of questions, their expected answers, incorrect answers we obtained by the poor choice of features and the way we corrected our model accordingly.

SNo	Feature(s) Used / Added	Question	Expected Answer	Obtained Answer
1	POS tags	What pros neutralize risk of migration?	These advantages offset the high stress, physical exertion costs, and other risks of the migration. Article : 109.txt	Apart from physiological adaptations, migration sometimes requires behavioural changes such as flying in flocks to reduce the energy used in migration or the risk of predation. Article : 109.txt
2	POS tags & Synonyms	Same	Same	Same as expected
3	POS tags & Synonyms	Who are the birds of prey?	These include many birds of prey such as vultures, eagles, and buzzards, but also storks. Article : 109.txt	Some birds of prey specialize on migrating waders. Article : 109.txt
4	POS tags, Named Entities & Synonyms	Same	Same	Same as above
5	POS tags, Named Entities, Synonyms & Lemmas	Same	Same	Same as expected

6	POS tags & Synonyms	When did major entries in Britian happen during the nineteenth century?	Bohemian waxwings Bombycilla garrulus show this unpredictable variation in annual numbers, with five major arrivals in Britain during the nineteenth century, but 18 between the years 1937 and 2000. Article : 109.txt	Same as expected
7	Synonyms & Lemmas	When were swans beak commented upon?	Swans have been marked with a nick on the beak since about 1560 in England. Article : 109.txt	Same as expected
8	Synonyms, lemmas, wordnet features, named entities & heads	shdkjsh djhskjdh	Out of scope for our system	****** Our System did not learn the knowledge required to answer this query ****** Article : unavailable

PENDING ISSUES

We are currently using reliable ways of obtaining answers to WHEN and WHO type questions while WHAT questions can still be improvised. Any feature which we are adding for WHAT is not always reliable and so, after rigorous analysis, we adopted a few reliable ways to answer WHAT questions.

POTENTIAL IMPROVEMENTS

As we all know that nothing in the world is ever perfect, even our system can be improved in a few dimensions. Our system currently uses features with weights assigned based on potential and rigorous analysis. However, we are currently injecting very little practical knowledge to our system. Also, as we are indexing each sentence, there's no scope to get discourse knowledge. If the system can be built in such a way that it follows several levels of granularity to drill down to the solution, discourse knowledge can be applied to get a bit more reliable results.

CONCLUSION

The Question-Answering System we proposed can be of real use in closed domain systems like exclusive Medicine fields, Specific research oriented studies etc. However, to extend it to open-domain systems, more data is required for training and more sophisticated technologies along with the Natural Language Processing approaches are needed to build a more reliable model. Our system offers a prototype for a Closed-Domain Question-Answering System that solely relies on NLP which can be used as a base to build models with interdisciplinary fields.