WSTA - Question Answering System

Name: ***Alisha Aneja****, Student ID:* ***872994****, Kaggle Team:* **hifive,** *Kaggle Username:* **aaneja17**

1. Introduction

The aim of this project is to develop a Question Answering system using the concepts learnt in the subject.

2. Method Description

Document Text Preprocessing: In order to find the relevant paragraphs for each question in the specified document, document tokenization and lemmatization is implemented. Lemmatization helps in shortening the vocabulary space, thus improving the size of the index. Stop word removal aids in reduction of noise in the documents as they are usually abundant and semantically not so relevant.

Paragraph Retrieval for given question: TF-IDF transforms the question and all the paragraphs in the document into vectors. Cosine Similarity calculates the similarity of the query vector with all the paragraph vectors and returns the most similar paragraph. The correct paragraph for the question is retrieved with an accuracy of 80% based on the validation set.

Sentence Retrieval from the paragraph: All the sentences for the paragraph retrieved are tokenized and the number of each word occuring in the sentence counted. The top 3 sentences which have the most similar tokens with the question, are retrieved.

Question Classification: The questions are classified in PERSON, PLACE, TIME, QUANTITY, PERCENT, ORGANIZATION or MISCELLANEOUS type questions on the basis of the words the system contains. For example, it checks the existence of words like ‘what’, ‘when’, ‘where’, ‘who’ in the question and the quantifiers like ‘much’, ‘many’, ‘few’ for QUANTITY, time words like ‘earliest’, ‘ago’, ‘next’, ‘before’ for TIME and similarly for other types.

Constituency Parsing and Named Entity Tagging: The three most similar sentences from the paragraph are parsed using Stanford Parser. It should be noted that the constituency parser is used instead of dependency parser. Reason for this is elaborated in the next section. Using this parser, the named entity tags of various constituents are compared with the type of question. For example, if the type of a question is PERSON, the constituents with PERSON tag in the sentences are returned. If the corresponding constituents are not found, the system returns the best matching substring from the most similar sentence in the paragraph as the answer.

3. Analysis

Comparative Analysis: Initially, the system was developed using dependency parsing and POS tagging. The latest system is developed using constituency parsing. This considerable improvement by using constituency parsing, as evident from Table 1.2, is because constituency parsing breaks a sentence into sub-phrases while dependency parsing connects words according to their relationship. Constituency parsing worked well in the cases where the questions are factoid based and their answers are nouns and sub-phrases from the sentence. For example, *‘Where do the Amtrak and Coaster trains primarily run?’*

This question would be easily classified as a LOCATION type question. The parser would find the constituent with named entity tag as PLACE which will return back 'San Diego'. Some more examples are:
'*How much of the earth is covered by oceans?*' -> QUANTITY
'When was Peterson born?' -> TIME
'What percentage did the crime rate in San Diego drop?' -> PERCENT
For all these questions, the system predicts correct answers. This is because constituency parsing predicts target phrases rather than words and captures the recursion in the sentence itself.

Moreover, using PERCENT and ORGANIZATION named entity tagging improved the accuracy considerably.

Error Analysis: *Unicode characters:* There is not one to one mapping for all unicode to ASCII conversions. Hence, the parser is unable to tag them when converted into ASCII by their named entity tag. For example, '*what is the Russian back-transliteration from Encyclopedia Britannica?'*. The answer to this question contains Russian characters. Hence, the system is not able to predict correct answer for this. Tools like MetaMap and ASCII Lexicon can be used to handle this in the future [2].

Reasoning Questions: Questions like *'What principles usually govern the Estonian orthography?'* cannot be classified into any particular category of question like TIME, PLACE etc. since they are the reasoning kind of questions. For this, a system could be developed to identify conjunctions/joining phrases in the sentences like 'because', 'as', 'that is why' to derive the semantic meaning from the sentence. This could be also done using neural networks, specifically Attention model [1]. Attention model computes the weighted sum of the values in the sentence, hence giving a selective summary of the information contained in the values. The question determines which values to focus on.

Unclear entity tags in case of DATE: Named entity tags returned by the parser sometimes returned DATE and other times returned NUMBER. One such example is '*79 ce*'. It is tagged as DATE by the parser, however *'1898', '2016'* etc. are marked NUMBER by the parser. Hence, some more filter conditions should be applied from the user end to get the correct answer.

4. Evaluation

The best system achieved an overall accuracy of 13% and an F-score of 16.7% (Table 1.1). The accuracy can be stretched till 30% if it is considered that the actual answer is the subset of the predicted answer. It should be noted that once the questions are categorized, PERCENT type questions are predicted with the highest accuracy amongst all other categories, which is 83.6% (Chart 1).

5. Conclusion

This system is developed using simple techniques of Natural Language Processing and achieves decent metrics. However, there is a lot of scope of enhancement in the system, by using some state-of-art techniques like Attention model, RNET and QANet, which can be tried on the dataset in the future.

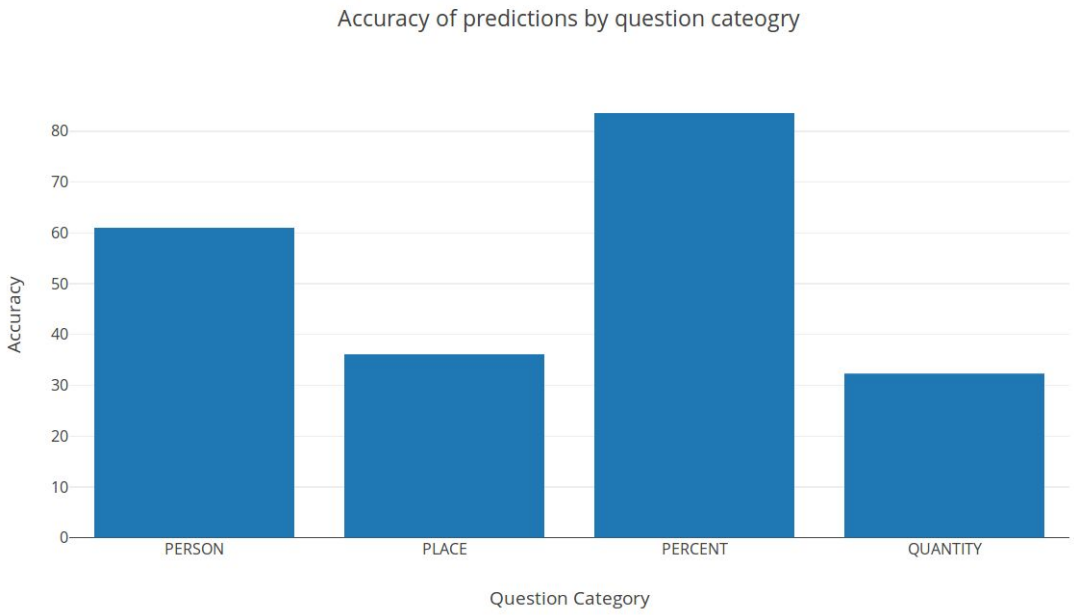

Chart 1: Accuracy of predictions by the question category

Accuracy	**F-score**
13%	16.7%

Table 1.1: Evaluation metrics for the best system

	Dependency parsing	**Constituency Parsing**
F1 score	12.5%	16.7%
Accuracy (if correct answer is subset of predicted answer)	30%	50.2%
Accuracy (if correct answer is exactly as predicted answer)	5%	13%

Table 1.2: Performance comparison between system using dependency parsing vs. system using constituency parsing

References

[1] Luong, T., Pham, H., & Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. http://doi.org/10.18653/v1/d15-1166

[2] Klensin, J. (2008). ASCII Escaping of Unicode Characters. http://doi.org/10.17487/rfc5137