QUESTION TYPE RECOGNITION AND CLASSIFICATION USING NATURAL LANGUAGE INPUT

A Project Proposal

Presented to

Dr. Duc Thanh Tran

Department of Computer Science

San Jose State University

In Partial Fulfillment

of the Requirements for the class

CS 297

By

Aishwarya Soni

November 2016

Abstract

Question type recognition is the most significant thing in the question answering type systems, for example, chat bots. When a user ask a question, the user expects a correct form of answer in reply. It is different from a document retrieval task where a document is retrieved using some keywords. To classify a question correctly, the system needs to understand to a much grained level, what type of question it is and what are the constraints associated with it. Question type classification drastically reduces the search space to retrieve an answer.

Problem Statement

The objective of this project is to create a supervised learning model that can recognize and classify a given question based upon its question type.

Proposed Solution

The model will be designed using a hierarchical classifier. The hierarchical classifier will take the input as a natural language input. The first classifier will classify the question into a "coarse" class and the second classifier will further classify it into a "fine" class. For example, given a question "*What is the most local place in Paris?"* The coarse classifier will produce the output as "LOC". The fine grained classifier will further classify the question to its subtype and it will produce the output as "*LOC:state*" For a baseline evaluation, the results of the evaluation matrix of the classifier has been compared with what has been obtained during by previous works. The evaluation matrix will include accuracy, precision, recall and the f1 measure score.

Work done for this semester

To train the classifier, first the features of a text is extracted and vectorized. The most important features that are considered is the tf-idf vectorizer, semantically related words, parts of speech (POS) tagging and Named Entity Recognition (NER). The features are extracted using custom classes which have inherited the *TfidfVectorizer* class. The built-in method *build_analyzer()* has been overridden to provide a custom functionality.

There are currently three custom classes which does the feature extraction process,

1. TagVectorizer- extract the POS tags
2. NERVectorizer- uses the in-built StanfordNERTagger provided in the nltk package to identify all the named entities
3. RelatedWordVectorizer- extracts the semantically related words from a bag of words

The model is trained using the extracted features and is then used to predict a question type given a text. Currently, SVM is being used as a classification algorithm. The model is trained and tested using TREC QA track dataset.

Challenges

Following are the main challenges that needs to be addressed in the work,

1. Identify the *k* semantic classes to predict.
2. Set the parameters to as optimized level as possible so that a better accuracy is achieved.
3. Finding the best classification algorithm for the model.
4. Make the model more scalable. For example, if multiple user concurrently submits a question, the application should handle it and predict the correct question type. For scalability, the plan is to use Apache Spark.
5. What kind of questions can be misclassified by the classifier? What can be the possible reason for that?

Timeline

Week 1: Oct. 9 - Oct. 16	Research supervised learning approaches
Week 2: Oct. 16 - Oct. 23	Establish baseline
Week 3: Oct. 23 - Oct. 30	Design the classifier
Week 4: Oct. 30 - Nov. 6	Obtain classification for a question type (Deliverable 1)
Week 5: Nov. 6 - Nov. 13	Evaluate performance
Week 6: Nov. 13 - Nov. 20	Research model optimizations
Week 7: Nov. 20 - Nov. 27	Improved implementation (Deliverable 2)
Week 8: Nov. 27 - Dec. 4	Evaluate performance
Week 9: Dec. 4 - Dec. 11	Draw Inferences and plan future work
Week 10: Dec. 11 - Dec. 18	Finalize and present project deck

Future Work

Improving the baseline by tuning the algorithm parameters. For large scale input, the code will be integrated with Apache Spark or some other big data tool. The model can be tested using some different classification algorithms to check if the accuracy improves or not. Currently the dataset used for testing is from the TREC QA track. The model needs to be tested upon the questions from the FAQ set.

Reference

[1] X. LI, and D. ROTH, (2006) “Learning question classifiers: the role of semantic information”, *Natural Language Engineering*, 12(3), pp. 229–249, June 2004. [Online]. Available: https://doi.org/10.1017/S1351324905003955