Data Analytics and Cognitive Computing Group Project

Detecting Duplicate Questions in Quora Dataset

Victoria Sardelli Orens Xhagolli

1. INTRODUCTION

For our project, we would like to develop a system that will determine if two given Quora questions are duplicates of each other. Towards the beginning of October, we began to work on our project and explore the capabilities of IBM Bluemix.

First, we researched previous attempts at solving this problem. We found many papers and resources regarding the determination of duplicate sentences. Two interesting sources in particular were found on Quora's website and Kaggle.com. On Quora's website, Nikhil Dandekar posted an article entitled "Semantic Question Matching with Deep Learning" in which he more formally defined the problem and briefly talked about three approaches that Quora engineers implemented as possible solutions. These approaches included a Long Short Term Memory (LSTM) with concatenation, an LSTM with distance and angle, and "Decomposable attention." The best accuracy they reached with those approaches was 87% by using either of the two LSTM Neural Networks. On Kaggle.com, a user named shubh24 briefly discussed six approaches that could be used to tackle the problem. Some of the approaches he mentioned were to use a simple Cosine/Jaccard Test, build a Decision Tree or Random Forest, or use the Explicit Semantic Analysis methodology. The resources we found helped us to learn about the wide range of strategies that exist to solve this problem, as well as the approaches that experts in the field are implementing and improving upon. Using the powerful Watson APIs that are available to us, we will try to replicate some of the techniques we found that are within our means to implement.

We will use Python for our project because it is supported by IBM, provides capabilities for fast prototyping (e.g. lack of types, list comprehensions, etc.), and has a rich set of libraries that we can use to clean and process data before and after we make calls to Watson's APIs. For example, by using Python's Natural Language Toolkit (NLTK) library, we can easily apply NLP techniques to our corpus such as stemming words to remove ambiguity, checking for synonyms, and using certain distance metrics to determine similairity between words. Spellchecking and finding synonyms is also made easier by the use of another Python library called Pyenchant.

IBM Bluemix provides a substantial list of cloud services, including those used for Natural Language Processing. Among these services are Naturak Language Classifier, Language Translator, Tone Analyzer, and Natural Language Understanding. They're fairly similar in the way they operate, but we found that the Natural Language Understanding and Language Translator services encapsulate many features of interest to us, and we decided to experiment with their APIs. The Natural Language Understanding (NLU) service is a set of REST APIs that allow you to analyze a corpus in a variety of ways. After conducting simple tests using some of the APIs provided, we decided that the entity recognition, keyword extraction, categorization, semantic role understanding, and concept recognition APIs were the most beneficial ones to use for our project. We experimented with these APIs to see what kind of information we could extract from example sentences and what kinds of issues we might come across.

The first two sentences we experimented with were "How can I be a good geologist?" and "What should I do to be a great geologist?" It is interesting to note that the first sentence returned information about the semantic roles of words in the sentence, such as the subject being "I" and the verb being the word "be." However, the second sentence did not return any results from the semantic roles API call. In addition, the two sentences did not return any information regarding the concepts in the sentence.

We compared this to the results returned when using a sentence about Star Wars, which returned a long list of concepts. The sentence was "How did Darth Vader fought Darth Maul in Star Wars Legends?" A potential issue we discovered when using this Star Wars question was how to handle sentences that used incorrect grammar or words. For instance, the person who asked the Star Wards question should have used the word "fight" instead of "fought." We decided that for now, in the early stages of development, we will not take into the consideration the tense of the verbs when determining the similarity of sentences. If the results of that strategy are unsatisfactory, then we will conduct more research and consider different ways to solve this problem.

Another example of a sentence we tried was "What is the difference between & and 'and'?" We were unable to glean any useful information from the results of the API calls. No concepts or semantic roles were returned, and the categories that were determined for this sentence included seemingly unrelated categories such as "cold and flu," "life insurance,"

and "divorce." We believe that this problem stems from the fact that the main interest of this sentence is the symbol "&," which Watson's APIs most likely do not know how to handle correctly. Another example of an incorrectly interpreted sentence involving symbols is the question "When do you use ffib7 instead of ãĄŮ?" The results of analyzing this sentence were similarly unhelpful. The issues with this latter sentence caused us to consider what we should do when we encounter words or characters from foreign languages within Quora questions, since the NLU APIs may not handle these as expected. A service that we considered using is the Language Translator service. Although it only works for a small set of languages, it is worth exploring to see if it can be used to aid our analyses. If we ecounter a sentence that has some or all of its words in a language other than English, we can attempt to translate those words into English before applying any techniques to determine sentence similarity.

We plan to continue exploring the capabilities of Watson and Python's Natural Language Processing libraries in an attempt to develop a solution to the problem of duplicate Quora questions.

2. REFERENCES