COMP9417 - Assignment 2

Avinash K. Gupta, Yuyang Shu, Maria Oei

June 4, 2017

1 Introduction

With over 100 millions of monthly visitor in Quora, it is inevitable that many people are asking similar questions. This has become an issue as user has to read through responses to many questions in order to find the best answer. The aim of this project is to implement algorithms to identify 2 similar questions which can help Quora in improving user experience by finding high quality answers to questions.

The 2 approaches used in this project were perceptron learning and LSTM. In the perceptron learning, we used 3 inputs which measured semantic similarity, word order and word overlaps between 2 sentences.

In LSTM, bla bla bla bla

We will see that for our case the performance of the 2 models were quite similar. However, there are other research done on LSTM where it has been shown that LSTM has the potential to performs well with enough training and with careful selection of initial weight.

What we used

2 Methodology

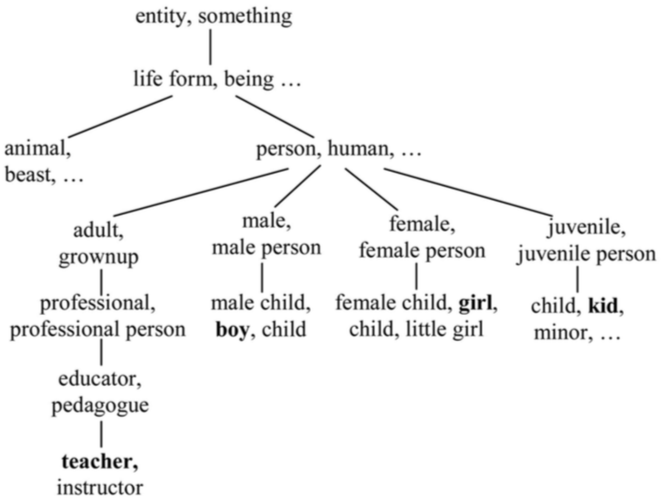
In this project, we used wordnet as the main lexical database. Using wordnet, we were able to extract the part-of-speech tags for a word as well as their synset (synonym set). This played an important role in determining the similarity of 2 words as we would see later.

2.1 Method 1 : Perceptron

The first model was a a simple perceptron with 2 inputs which were based on [insert reference to paper here]. The inputs chosen measured the semantic similarity and the word order information of the 2 sentences.

2.1.1 Words Similarity

A **path length** between 2 words is the number of synsets we visit from one word to another. For example, in the figure below, to get from boy to girl we have to visit boy - male - person - female - girl. Therefore, the path length of 'boy' and 'girl' is 4. 'Person' is called the subsumer of 'boy' and 'girl'. If there are more than 1 path, we will consider the shortest path and the corresponding subsumer is called the **lowest subsumer**.



Let l denote the shortest path and h denote the depth of the lowest subsumer. The similarity between 2 words w_1 and w_2 is therefore measured by

$$s(w_1, w_2) = f_1(l)f_2(h)$$

where $\alpha, \beta \in [0, 1]$ and

$$f_1(l) = e^{-\alpha l}$$
$$f_2(h) = \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}$$

2.1.2 Semantic Similarity

Let T_1 and T_2 be the 2 sentences and T is a set of distinct words in T1 and T2. For each sentence, we will calculate the vector s_k which the same length as T. For each word w_i in T, we assign 1 to the corresponding element in s_k if w_i is in T_k and μ_i otherwise. μ_i is the similarity score between w_i and the most similar word in T_k calculated based on the similarity score above. Once s_1 and s_2 are calculated, the overall semantic similarity score is calculated by

$$S_s = \frac{s_1.s_2}{||s_1||.||s2||}$$

2.1.3 Word Order

Similar to the semantic similarity, we calculate the vectors r_1 and r_2 for each of the sentences. For each word w_i in T, we set the i^{th} element in r_k to equal to the position of w_i in T_k. If w_i is not in T_k then we find the most similar word in T_k and assign the position of that word instead.

For both word order and semantic similarity measure, we define a threshold for the case where w_i is not in T_k. If the similarity score between w_i and the most similar word is less than the threshold, 0 will be assigned instead.

For the word order, the overall score is calculated by

$$S_r = 1 - \frac{||r_1 - r_2||}{||r_1 + r_2||}$$

2.2 Method 2 : LSTM - RNN

3 Result

4 Discussion

4.1 Conclusion

4.2 Limitation & Improvement