Independent Study with LIL Lab

Application Project

The aim of the project is to classify an English translation of a Mandarin sentence as being translated by a machine or by a human. From a quick look at the dataset, each example has 3 sentences, one being the sentence in Mandarin that is being translated, a human translation of the sentence, a candidate translation of the sentence that is either translated by a machine or by a human, and a score rating the quality of the translation. Given that this is a classification problem, the first algorithms I think of are SVMs with kernels designed for sentences features or LSTMs. Given the number of sentences and the fact that we are checking the quality of a sentence rather than looking for concepts in sentence, I made an assumption that the data would be extremely high dimensional. Thus, due to the low number of examples in the train and test set, I didn’t want to overfit the train set. Given the property of SVMs that prevents overfitting even with high dimensional data, I chose to model this data set with an SVM. Hence, I looked for research papers that delve into using SVMs for text classification. I found two research papers: First was *“Text Categorization with Support Vector Machines: Learning with Many Relevant Features”* by Professor Thorsten Joachims in 1998. This involved using tokenization and word lemmatization with an SVM for text classification. The second was *“A Machine Learning Method to Distinguish Machine Translation from Human Translation”* by Yitong Li, Rui Wang, and Hai Zhai. This one was based on the previous research using SVMs for text classification including the first research mentioned. However, the authors of this research used the Stanford parser, and designed features using the information provided. I emailed the authors of this paper to see the code for this paper, but they did not reply. Unfortunately, developing the features was a process that would take a lot more time to do given that I would have to figure it out from scratch, and the time limitations prevented me from attempting to do so. Hence, I decided to implement the version described in Professor Thorsten’s Paper. For all the three different sentences provided for each data point, I first use a word lemmatization function. Then I tokenized the data points, using a model to vectorize sentences. This was fitted to the overall data set, so that the dimensions of the train and test sets match. I then used a hard margin SVM on the data. This yielded me a success rate of 78.1609% on the test set. Seeing this, given the time constraints, I decided to stop here for this project.

Attached below are links to the papers:

1. T. Joachims, Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Proceedings of the European Conference on Machine Learning (ECML), Springer, 1998. <https://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf>
2. Yitong Li, Rui Wang, Hai Zhang, *“A Machine Learning Method to Distinguish Machine Translation from Human Translation”*, PACL 2015.  
   <http://aclweb.org/anthology/Y15-2041>