Dariia Dragunova
27 October 2022
CIS 4526

ML Midterm: Systems Description

Features Description

For this project, I tried to identify features that would help tell if two sentences were similar or not. Most of the features considered the structural aspect of sentences such as their length, number of identical words, number of n-gram overlaps and more. By looking at sentences in the training and development datasets, I tried to identify differences between the sentences and how I could find those differences in the algorithm. Some things that stood out were the capitalized words and numbers that appeared in a lot of sentences. Below is the list of features that were used for my model along with their descriptions.

1. Word Count Difference
    a. This feature compares the number of words in each sentence and returns the difference
2. Word Overlap
    a. This feature compares the words in each sentence and returns the number of words that are the same in both sentences
2. Word Union
    a. This feature counts the number of unique words in two sentences
3. 3 Gram Overlap
    a. This features calculates three gram overlaps in each sentence and returns the number of 3-gram-overlaps that are the same in both sentences
4. 3 Gram Union
    a. This feature counts the number of unique 3-grams in both sentences
5. 2 Gram Overlap
    a. This features calculates three gram overlaps in each sentence and returns the number of 2-gram-overlaps that are the same in both sentences
6. 2 Gram Union
    a. This feature counts the number of unique 2-grams in both sentences
7. Numbers Overlap
    a. This features counts occurrence of numbers in a sentence and returns the difference between sentences
8. Capitalized Words Overlap
    a. This features counts occurrence of numbers in a sentence and returns the difference between sentences
9. Cosine Similarity Score
    a. This feature calculates the cosine similarity score for two sentences

Data Preprocessing

1. Identifying bad lines
    a. When opening the files, I had trouble processing some lines because of bad formatting. Some lines had strings in the column for ‘gold_label’ as well as extra quotation marks in the sentences themselves.
2. Working with sentences
    b. Processing the strings was a very important aspect of this project. All the features dealt with identifying structural components of sentences, thus it was important to perform the following operations on sentences.
        i. Lowercase all words in a sentence
        ii. Tokenize the words
        iii. Remove the stopwords
        iv. Lemmatize the words (reduce the different forms of a word to one single form)

Libraries

1. NLTK
    a. NLTK or Natural Language Toolkit is a Python suite of libraries and programs for symbolic and statistical natural language processing in English
    b. I used this library for data preprocessing and feature extraction. Useful functions helped me tokenize, lemmatize and perform other manipulations with sentences.
2. Pandas
    a. Pandas is a Python library written for data manipulation and analysis.
    b. Used Pandas to load the data, create data frames and perform manipulations on the data.
3. Sklearn
    a. Sklearn is a Python library that provides a selection of efficient tools for machine learning and statistical modeling.
    b. Used this library to create and experiment with different models.
4. Re
    a. Re is a module of regular expression matching operations.
    b. Used Re library to dissect sentences and find useful features in them such as capital letters and numbers.
5. Math
    a. Math is a module that provides access to the mathematical functions defined by the C standard.
    b. Used math to perform basic mathematical operations as well as matrix manipulations.
6. Seaborn and Matplotlib
    a. Used these two libraries to visualize the data

Algorithms

I tested four different models for the this task.

1. Linear Regression
    a. Tried using Linear Regression as a starting point. This algorithm did not work well for this task outputting 46% training score.
2. Decision Tree
    a. Similarly to Linear Regression, this model did not work well for NLP outputting training score of around 45%.
3. Logistic Regression
    a. Expected this model to work better since it's using a non-linear function to approximate the decision boundary. Logistic Regression improved the accuracy to around 55%.
4. SVM
    a. Linear SVM
        i. Tried by using linear SVM and got accuracy of 50%.
    b. RBF SVM
        i. Tried using RBF and got better scores than Linear SVM but the scores weren't consistent.
    c. Kernel SVM
        i. After visualisizing the features with the gold label, it became obvious that it would be very challenging to linearly separate data and we need a model that would support a non-linear boundary.
        ii. Kernel SVM has a degree parameter that controls the flexibility of the decision boundary. By modifying this parameter, I was able to increase the accuracy of the model.
        iii. Eventually, the optimal model was based on a Kernel SVM with degree 8 giving me accuracy of 65%.

Experience and Lessons

a. Working with a large data set
    i. I have never worked with a dataset this large. It was a new experience for me and taught me a lot about data pre-processing. It also showed me how long it may take to run a model on larger data sets. For some of the models, training and testing would take up to 40 minutes.
b. Trying out different algorithms and seeing them work
    i. Before starting this project, I thought the process was going to be very straightforward. I have learned all these algorithms and models in class and knew about the benefits of using one model over the other. After experimenting with Linear Regression, Logistic Regression and Linear SVM I was disappointed to see my accuracy score still be very low. I decided to visualize the data and found

no linearly separable data. I remembered the kernel SVM model from class to solve this problem and it ended up working much better for this kind of data. Working with these models hands-on gave me a larger understanding of why people choose one model over another.

c. Reaizing why Neural Networks exist
    i. Experimenting with different models and finetuning their parameters has been able to increase the accuracy score for this NLP task but nothing compares to Neaural Networks. Compared to 58-65% range that I was able to get with my models, NN were able to achieve 88-90%. Comparing these results, made me understand why they were developed and are so widely used nowadays. Although we weren't allowed to use Deep Learning in this project, it gave me greater appreciation for it.
d. Realizing some features are not as important as others
    i. After creating all the features and running different models I started to understand that some features were more important than others. For example, the word union feature that I developed had little to no effect for the model. It took visualizing and playing with removing/adding some features to learn which ones were the most significant to use in the algorithm.

Resources

- https://cla2019.github.io/paraphrases_scikit_numpy.pdf
- https://towardsdatascience.com/top-10-binary-classification-algorithms-a-beginners-guide-feeacbd7a3e2