Kaggle Competition
Predicting word complexity
Machine Learning Course

2st Semester of 2023-2024

Turis Gavriil-Vlad
gavriil-vlad.turis@s.unibuc.ro

Ilie Octavian Tudor
octavian-tudor.ilie@s.unibuc.ro

Abstract

This document is a report of a Kaggle competition that is a project for the machine learning course. Our team is formed by: Turis Gavriil-Vlad and Ilie Octavian Tudor. We needed to predict the word complexity for this competition using a regression model. We had 8633 words for the train data and they were in different languages : catalan, english, filipino, french, german, italian, japanese, portuguese, sinhala, spanish. The test data was formed by 5623 words and we needed to predict the complexity column. To evaluate our progression we used: Coefficient of Determination, and Person Correlation Coefficent.

1 Introduction

- In this project, we aim to develop a machine-learning model to predict the complexity of a given word within a sentence

- Turis Sef: Researched and identified a comprehensive set of features related to word complexity, from the most general to the most detailed and performed grid search to find the best parameters for the model.

  Tudor Sef: Analyzed and selected the best features for predicting word complexity by testing different combinations of features.

- Evaluated the impact of different feature sets on model performance.
  -Evaluated the impact of different parameters for the model performance
  -Evaluated the performance of K-Nearest Neighbors (KNN) and Random Forest Regressor models .
  -Analyzed the importance of each feature in the obtained score.
  Tested the effects of including too many features versus too few features.

- Searching for general features Testing KNN vs Linear Regressor vs MLP vs RandomForest+Regressor with the general obtained features. Searching for more complex features (which use libraries like wordnet , spacy , BERT). Testing the Random Forest Regressor model with all obtained features. Testing the Random Forest Regressor model with an optimal combination of features: Using GridSearch to find the best parameters

- -Discover all the steps involved in creating an artificial intelligence model.
  -Gain familiarity with the Kaggle platform and its tools for data science competitions and project collaboration.
  -Train our analysis and research skills in solving a given task.
  -Evaluate our interest in the field of artificial intelligence to see if it aligns with our career goals.

- Studies on how Random Forest works and its effectiveness
  Research on the capabilities and applications of SpaCy, WordNet, BERT and RoBERT libraries
  Exploration the embedding technique, which transforms words into numerical vectors to capture semantic meaning
  Research on the parameters of RandomForestRegressor, such as the number of trees, maximum depth, and splitting criteria

2 Approach

- Github Link: https://github.com/Tudorr02/IA

- For coding we used Python and the IDE used was Visual Studio Code and also the Kaggle platform.

- The training process takes now 5 seconds, but in the past when we tried various features we had waited several minutes.

- We used a regression model named Random Forrest Regression, Linear Regression, Spacy library, and WordNet. Also, we tried K-Neibgbour Regression, MLP Regressor and some transformer models like BERT and RoBERTa, for word embeddings but with no success.

- We added features and we selected them based on their importance and their effectiveness, and also we tried to not overfit the model. Another trick we used was preloading the spaCy language models and storing them into a dictionary. This approach significantly optimize data processing.

- After testing different combinations of features types of features, we finally choose 12 of them. We will list them with a brief description: Word Length :
  Calculates the length of the word.
  Vowel Count : Counts the number of vowels in the word.
  Consonant Count : Counts the number of consonants in the word.
  Part of Speech : Retrieves the part of speech tag for the word.
  Word Frequency : Computes the frequency of the word from an external corpus.
  Synsets Count : Counts the number of synsets (sets of cognitive synonyms) for the word in WordNet.
  Synsets Depth : Determines the maximum depth of the synsets in the WordNet hierarchy.
  Hypernyms Count : Counts the number of hypernyms (more general terms) for the word in WordNet.
  Hyponyms Count : Counts the number of hyponyms (more specific terms) for the word in WordNet.
  Root Distance : Calculates the distance of the word from the root of its dependency tree in the sentence.
  Syllable Count : Determines the number of syllables in the word.
  Capitalization Ratio : Calculates the ratio of capital letters to the total letters in the word.

3 Conclusions and Future Work

- From our point of view, we could have done differently plenty of things. For instance, the communication in the team could have been better. We certainly, get along very well, but in the heat of the moment and with other considerable tasks in this period on our minds, both at work and as well as at university with the examination session and other projects. I tend to think that, we could enhance our communication overall and maintain a calmer environment.

- Definitely, by adding more effective features and doing a more elaborate grid search for the newly added features.

- The project itself, was extremely engaging, however the fact that we had only five submissions daily and other projects and exams on our head, made it less enjoyable. On top of that, the concept of the project was very interesting and highly relevant nowadays.

- By doing this project, we learned considerably new pieces of information about how a proper machine-learning project is done and about. Moreover, we learned about relevant aspects that make a word more complex, and how to use artificial intelligence concepts to solve a given task.

- From our point of view, an Image classifier and a football match score prediction would be nice.

- More daily submissions and more train data.