Introduction to NLP
Interim Report

Team Name : Edge
Team No. : 43
Project Title : Measure Text Fluency (3)
Team Members : Sankalp Thakur (2021201042)
Shravan Sharma (2021201058)
Vineet Agrawal (2021201049)

1. Introduction

The focus of our project is to measure text fluency, which has become increasingly important due to the abundance of textual content in the world. Text fluency is a measure of the quality, readability, accuracy, and comprehensibility of a piece of text. As automated sources are prone to language errors, measuring text fluency is crucial in ensuring that the information conveyed is in a desirable format. This is a challenging task as it involves assessing various criteria such as grammar, style, and word choice. Our approach involves a multi-class classification problem, classifying a piece of text as either Not Fluent, Neutral, or Fluent. Our work has involved conducting a literature survey to understand previous approaches, implementing a baseline, conducting experiments using supervised machine learning algorithms, and analyzing the results in detail

2. Dataset

The Microsoft Compression Dataset was utilized in our work, comprising 22956 data samples. Each sample consists of a source , shortening , AverageGrammar and AverageMeaning. The dataset is actually based on compression of source sentences into shorter versions using 4 different techniques. These shorter versions were than rated manually be 3-5 judges on the basis of meaning and grammar. We use these shortened sentences and the two scores. Using the two scores we calculated the fluency score. We took weighted average of these scores, assigning 0.2 weight to meaning and 0.8 to grammar. Then rounded the score to nearest integer. Anything above 3 is discarded as invalid and we get 3 classes : 1 -> Non Fluent , 2-> neutral and 3 -> fluent.

3. Methodology

For the baseline model we use simple text to sequence to vectorize the data. All the words in train vocab are given an integer id and sentences are encoded according to this mapping to make every data point of equal size we pre pad every element of the dataset with 0 such that every encoding has the size of the largest element in the dataset. Note that an element in dataset can have multiple sentences.

For classification we use multiclass logistic regression and SVM.

Class Balancing:

1.RandomOverSampler : Random oversampling involves randomly duplicating examples from the minority class and adding them to the training dataset.

2.SMOTE : SMOTE or Synthetic Minority Oversampling Technique is an oversampling technique but SMOTE works differently than typical oversampling. While it increases the number of data, it does not give any new information or variation to the machine learning model.

3. BorderlineSMOTE : This algorithm is a variant of the original SMOTE algorithm. Borderline samples will be detected and used to generate new synthetic samples.

4. ADASYN : This method is similar to SMOTE but it generates different number of samples depending on an estimate of the local distribution of the class to be oversampled

4. Baseline Analysis

Logistic regression

	accuracy	precision	recall	F1
train	0.849	0.50	0.35	0.34
test	0.852	0.48	0.35	0.34
RandomOverSampler(train)	0.468	0.46	0.47	0.46
RandomOverSampler(test)	0.446	0.35	0.38	0.28
SMOTE(train)	0.488	0.49	0.49	0.48
SMOTE(test)	0.570	0.36	0.37	0.32
BorderlineSMOTE (train)	0.68	0.67	0.68	0.67
BorderlineSMOTE (test)	0.60	0.35	0.36	0.34
ADASYN(train)	0.48	0.49	0.49	0.48
ADASYN(test)	0.58	0.36	0.38	0.32

SVM

	accuracy	precision	recall	F1
train	0.856	0.53	0.37	0.37
test	0.853	0.34	0.35	0.34
RandomOverSampler(train)	0.367	0.49	0.49	0.48
RandomOverSampler(test)	0.839	0.33	0.29	0.29
SMOTE(train)	0.37	0.67	0.68	.067
SMOTE(test)	0.84	0.34	0.35	0.30
BorderlineSMOTE (train)	0.66	0.49	0.49	0.48
BorderlineSMOTE (test)	0.82	0.34	0.33	0.31
ADASYN(train)	0.404	0.60	0.41	0.31
ADASYN(test)	0.836	0.34	0.36	0.31

We performed experiments using multiple models for classification and used different sample to reduce the bias created due to data imbalance. We see really high accuracy in simple training data because the predictions are overwhelmingly of the largest class which is about 85 percent of the data. SO if the model only predicts that majority class it would get 85 % accuracy hence it is not a good measure for comparison of models.We can see that there is not much improvement when using sampling data to reduce unbalance. The models fail to capture the details properly.

5. **Code**

   The code can be accessed at **github**.

6. **Further plan**

   We plan to use some other techniques to represent the data like glove embeddings,bert embedding. Also for classification we will be using tree based models and models based on neural networks.We will also experiment with key metrics for text fluency such as ROUGE-S, ROUGE-L, SLOR, n-gram overlap for RNN and LSTM language models