Improving Writing Assistance at JetBrains AI - test task

November 3, 2024

Data

The `data1.csv` was taken from Robert Heckendorn's List of Hard to Spell Words: http://marvin.cs.uidaho.edu/misspell.html. The `data.csv` was taken from Kaggle: https://www.kaggle.com/datasets/bittlingmayer/spelling

In order to obtain a `.csv` file (from the second dataset), with which it will be comfortable to work, I created a script : `get_csv_script.py`

Used models

1. **PySpellChecker and Aspell** rely on a predefined dictionaries of correctly spelled words. To correct a word misspelling these models count the Levenstein distances (for the misspelled word and the words from models' predefined dictionaries).After this, algorithms pick the closest words from the dictionaries, and choose the word which is the most frequent in the language.
2. **TextBlob** relies on a probabilistic model for spell correction. It uses a predefined dictionary of common words and considers the probability of each word based on its frequency in a large corpus of text.
3. **BERT** (Bidirectional Encoder Representations from Transformers) is a transformer-based model, which can also be used for spelling correction.

Metric 1 - percentage of correctly spelled words

Table 1: Correctly Predicted Words on Kaggle Dataset

Model	Percentage of Correct Predictions
Aspell	79.39%
PySpellChecker	73.77%
TextBlob	61.83%
BERT	0.90%

Table 1: Correctly Predicted Words on Kaggle Dataset

Table 2: Correctly Predicted Words from Robert Heckendorn's List of Hard to Spell Words

Model	Percentage of Correct Predictions
Aspell	58.90%
PySpellChecker	47.98%
TextBlob	34.35%
BERT	0.33%

Table 2: Correctly Predicted Words from Robert Heckendorn's List of Hard to Spell Words

Metric 2 - Mean Levenshtein Distance

Table 1: Mean Levenshtein Distance from Kaggle Dataset

Algorithm	Mean Levenshtein Distance
Aspell	0.5316
PySpellChecker	0.7255
TextBlob	0.7817
BERT	7.5980

Table 3: Mean Levenshtein Distance from Kaggle Dataset

Table 2: Mean Levenshtein Distance from Robert Heckendorn's List of Hard to Spell Words

Algorithm	Mean Levenshtein Distance
Aspell	1.4203
PySpellChecker	2.1297
TextBlob	1.9080
BERT	7.1337

Table 4: Mean Levenshtein Distance from Robert Heckendorn's List of Hard to Spell Words

Results

1. As we can see, for each of the two metrics, the best results shows an Aspell model, and the worst are definetely shown by BERT.
2. In general,observing the Levenshtein distance results, we conclude that words from Robert Heckendorn's List are much more difficult to deal with than from kaggle for Aspell, PySpellChecker and TextBlob. Interestingly, but for BERT the mean Levenshtein distance for these two datasets is almost the same.

Conclusion and improvements for BERT

1. It was not surprising, that even though BERT is a much more powerful engine in NLP, it deals with misspellings worse, than dictionary based models. This occurs because these two datasets consist of isolated words with no contextual embedding. And I am pretty sure that BERT will perform great when it will have this contextual embedding for misspelled words, because it is a LLM.
2. Also, it seems to me, that TextBlob deals worse with words, that are not frequent in language, because it's correction is based mainly on probability. So in these cases I think it will be better to use Aspell.
3. Finally: Aspell, PySpellChcker, TextBlob deal quiet good with misspellings in isolated words. BERT deals poorly with isolated words, but certainly will perform great having a contextual embedding for misspelled words.

Reproducing the results

In order to reproduce the results open `exploration.ipynb` and start compiling cell by cell.