CS 626 Project - Text Normalization

Changam Dileep Kumar Reddy 170050080
B.Nikhil 170050099

December 12, 2020

Problem Statement

Many speech and language applications, including text-to-speech synthesis (TTS) and automatic speech recognition (ASR), require text to be converted from written expressions into appropriate "spoken" forms. This is a process known as text normalization.

Input: sequence of tokens.
Output: spoken form of the tokens.

Example: 12:47 to "twelve forty-seven" and 6 ft into "six feet".

Methodology

Instead of Normalizing whole text into spoken forms, we classify each token into appropriate semiotic class and then apply rules to classify each token based on its class.
There are 16 semiotic class categories

PLAIN, PUNCT, DATE, LETTERS, CARDINAL, VERBATIM, DECIMAL, MEASURE, MONEY, ORDINAL, TIME, ELECTRONIC, DIGIT, FRACTION, TELEPHONE, ADDRESS.

Now our task is just a simple classification problem. We have used two methods for classification

- Neural Network approach for classification
- Traditional Machine learning approach - **XGBoost**

Dataset

The Dataset consists of 5 features:

- `sentence_id` - The index of the sentence this token belongs to.
- `token_id` - The index of the position within the sentence this token belongs to.
- `before` - Original text.
- `after` - Normalized text.
- `class` - The class to which this token belongs to.

sentence_id	token_id	class	before	after
0	0	PLAIN	Brillantaisia	Brillantaisia
0	1	PLAIN	is	is
0	2	PLAIN	a	a
0	3	PLAIN	genus	genus
0	4	PLAIN	of	of
0	5	PLAIN	plant	plant
0	6	PLAIN	in	in
0	7	PLAIN	family	family
0	8	PLAIN	Acanthaceae	Acanthaceae
0	9	PUNCT	.	.
1	0	DATE	2006	two thousand six
1	1	LETTERS	IUCN	i u c n
1	2	PLAIN	Red	Red
1	3	PLAIN	List	List
1	4	PLAIN	of	of
1	5	PLAIN	Threatened	Threatened
1	6	PLAIN	Species	Species
1	7	PUNCT	.	.
2	0	PLAIN	Circa	Circa

Figure 1: Dataset

The classes define the kind of normalization that should be applied on the token for a correct result. For instance, not every 20 should be converted in the same way. If the token is of the CARDINAL class, the correct result is twenty. However, if the class is ORDINAL, the output should be twentieth, and if the class is DIGIT, the correct output is two o.

class	count
PLAIN	7353693
PUNCT	1880507
DATE	258348
LETTERS	152795
CARDINAL	133744
VERBATIM	78108
MEASURE	14783
ORDINAL	12703
DECIMAL	9821
MONEY	6128
DIGIT	5442
ELECTRONIC	5162
TELEPHONE	4024
TIME	1465
FRACTION	1196
ADDRESS	522

We can clearly observe that >75% of data is of PLAIN class. Some classes like ADDRESS, FRACTION and TIME are of negligible percentages.

Data Preprocessing and Features

- Data set contains approximately 10 million tokens, considering the Time and memory constraints, we have chosen to pad/truncate each token to fixed length of 10.
- Now, each token is character-level encoded using their ASCII codes. We didn't use one-hot encoding because our task mainly requires us to identify whether the character is alphabet, numerical or special character and we know that ASCII codes for each class of characters are placed side by side.

  For Example: day is encoded as $[100, 97, 121, 0, 0, 0, 0, 0, 0, 0]$.
- Feature set of current token is constructed using encoded token of previous, current and next token. This is chosen to capture context from the sentence. Therefore, feature set is of length 30.

Neural Network approach for classification

Since we were using **LSTM** throughout the course. We started with basic LSTM architecture. Using the feature set mentioned in the above section, the best result we could get was an accuracy of approximately 93%. Also, this approach hardly classify any token as ADDRESS, FRACTION, TIME, TELEPHONIC, ELECTRONIC as they are less frequently appearing classes.
As this in not in line with the performance we are after and considering the distribution of classes in the dataset, we opted to pursue traditional Machine learning methods instead of Neural Networks.

Traditional Machine learning approach - XGBoost

- Feature set context from previous token is not much useful, As most classification can be done only using current token's features. This might be the reason for failure of LSTM.
- We thought decision tress might perform well because, just by looking at features we can make decisions to seperate most of the classes.
- Therefore, we used **XGBoost** - Extreme Gradient Boosting, a sequential ensemble learning algorithm, which is quite efficient and fast.
- We have used the same input, output from LSTM.

Results

Confusion Matrix:

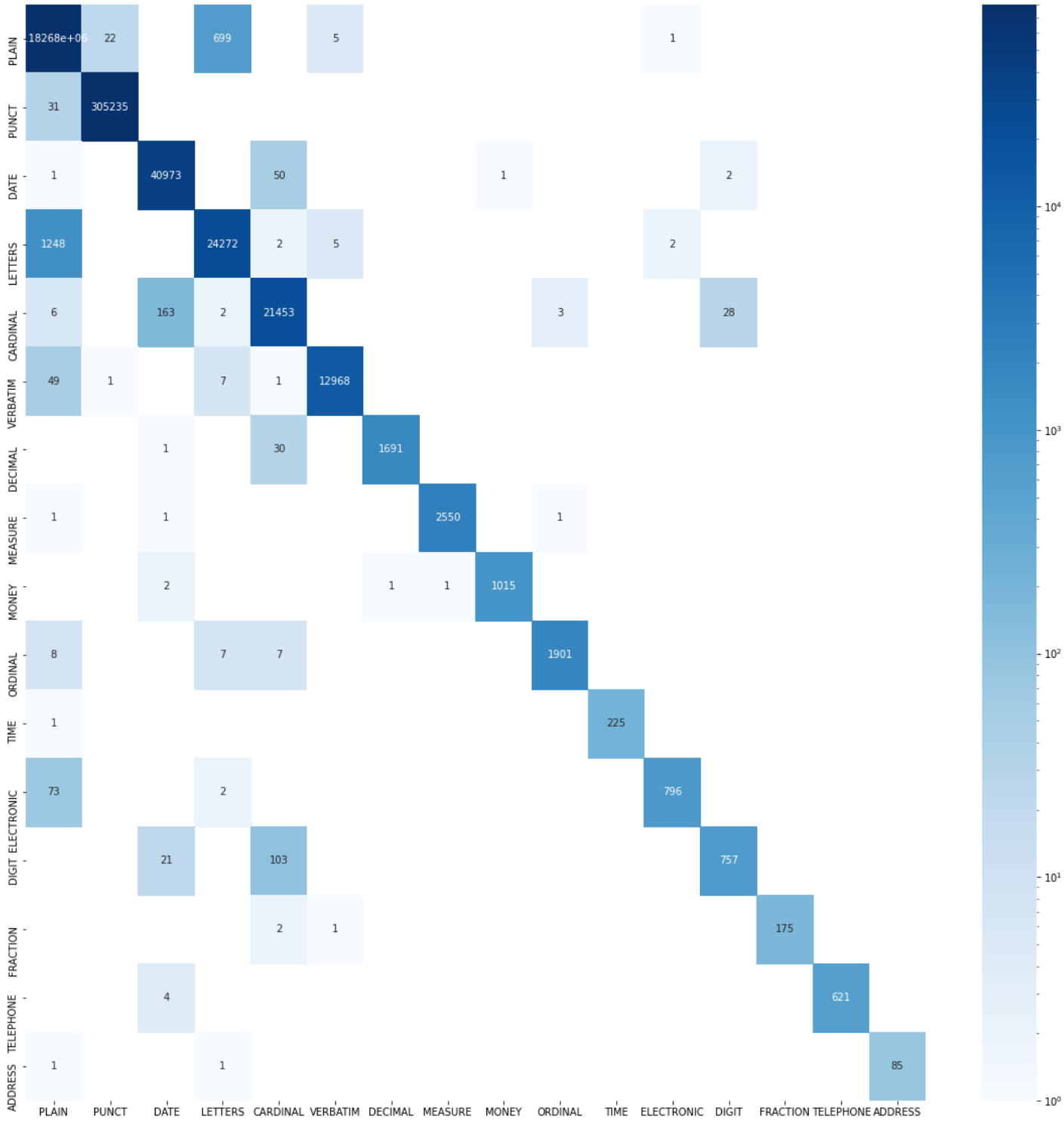

Figure 2: Confusion matrix with logarithmic heatmap

We can observe from the class-wise accuracy table and confusion matrix that the following classes are often misclassified.

- DIGIT and CARDINAL
- ELECTRONIC and PLAIN
- LETTER and PLAIN

We have listed the probable reasons for the misclassification of above classes in the error analysis section.

Class wise accuracy:

class	Accuracy
PLAIN	99.94
PUNCT	99.98
DATE	99.87
LETTERS	95.15
CARDINAL	99.06
VERBATIM	99.56
DECIMAL	98.45
MEASURE	99.92
MONEY	99.52
ORDINAL	98.88
TIME	98.18
ELECTRONIC	91.10
DIGIT	85.58
FRACTION	98.91
TELEPHONE	99.68
ADDRESS	98.86

Error Analysis

This section contains error analysis of most commonly misclassified classes.

0.1 LETTER and PLAIN

We know that even though words like AIDS, RADAR, PIN, IMAX, ASAP are abbreviations they are to be classified as PLAIN and not LETTERS. Our model is able to classify AIDS correctly as PLAIN because of frequent occurence of AIDS in the DATASET. However, unknown/rare words like IMAX and SCUBA are incorrectly classified.

0.2 ELECTRONIC and PLAIN

The misclassification between ELECTRONIC and PLAIN class is because we are limiting the token size to be 10, Therefore long website(without http/https) are truncated and imitate like PLAIN class.

For Example: "TheHuffingtonPost.com" will be truncated as "TheHuffing" and will be tagged as PLAIN.

0.3 DATE, CARDINAL, TELEPHONE and DIGIT

Since we are using character level encoding as features and decision trees for classification. Our model learns so hard that any token of length 4 starting with 19 or 20 is classified as year.

For example: In the sentence "more than 1900 soldiers were injured during World War I.",
1900 is wrongly tagged as year.

Similarly if the cardinal starts with "9"s then it will be misclassified as TELEPHONE.

Future Work

- Our Dataset contains already grouped words under the same token. But identifying token boundaries itself is a manual task. Therefore, Identifying token boundaries can be future work for this project.
- We can include more number of classes for,
  - Chemical Formulas like NO - nitrogen monooxide
  - Short form notation for words like "Contd", "i.e.,", "etc".