Training Models for Part-of-Speech Tagging

Katherine Ann Sick
Dep. of Computer Science and Linguistics
University of Illinois at Urbana-Champaign
kasick2@illinois.edu

Abstract

This paper follows the development of part-of-speech tagging classifiers. This is a fairly common supervised problem that is used to produce and study linguistic annotation. This study trains a bi-directional LSTM model and the Naive Bayes model in order to correctly identify the part of speech of each word in a sentence. The corpus that was used to train these models is a Universal Dependencies dataset containing a human-annotated POS tag for each word in a large database of sentences. Looking at the evaluation for each of the trained models revealed that the Naive Bayes model resulted in a much higher performance than the bi-directional LSTM model. However, when this model was applied to a new database, there were noticeable incorrectly labeled words throughout the database. We conclude this study by suggesting improvements that could improve accuracy and discussing the implications of creating POS tagging models.

1 Introduction

Predicting and understanding the parts-of-speech of sentences allows models to better understand grammar and sentence structure. In training models to do this, we can obtain more accurate analysis texts and corpora which can be highly useful for natural language processing or machine learning tasks. However, there are still multiple issues that can be encountered in doing this task. Word ambiguity and uncommon words both provide substantial challenges to having correct analysis from our models.

In this paper, we experiment with training multiple models to find the most optimal one to use. This way we will have a standard to compare to and will have more accurate predictions when running the model on a separate dataset. The goal in doing this is to try and create an accurate prediction model that can identify the parts of speech for any given word, in any given sentence.

After training, evaluating, and choosing the best model to move forward, we were able to apply it to another large dataset containing sentences gathered from newspapers. In doing this, we were able to confirm that the model was correctly trained and could be used in multiple contexts, and with multiple types of data.

2 Data Gathering and Processing

Two main corpora were utilized in this study. First, a Universal Dependencies database that was used to train our models, and second, a database of sentences taken from newspapers that we tested our trained models on.

To successfully use these corpora in our study, we had to clean and correctly format them. For the UD database, everything was converted to a CSV file from a text file, and we changed the format of the files to be an array of words, followed by an array of its POS tags, rather than each line being for one word. Then, after training the models using this data, we updated our second database by having it follow the same form as our UD database, and then converting it to a CSV file as well.

Pre-processing our data in this way gives us a clear way to see how our data is organized, and also lets us easily use it with our models.

3 Model Training

In this study, we train two different models to predict the parts of speech tags of words.

The first model that was trained was a bi-directional LSTM model which is a type of neural network that I had a bit of previous experience with in a past project. This type of model can be beneficial because it can take into account the context of the surrounding sentence for each tag. The way the bi-directional LSTM works is by processing each sentence that is inputted both forwards and backward so that it can train based on not only the

words that come before a part of a sentence but also the words that come after. This helps it to get a more accurate prediction of what the correct part of speech tag is.

Before deciding to use this model, it was originally planned to use a logistic regression algorithm instead, but I ran into multiple problems while trying to implement this algorithm. One of the biggest issues with this was with the size of the dataset, as there were multiple memory and storage errors relating to this. Upon switching to our final bi-directional LSTM model, however, I did not have any more issues in this area.

The second model that was trained was the Naive Bayes model. For this study, I used sklearn's Naive Bayes model (MultinomialNB) which tagged words by treating it as a word classification problem. To do this, we first used a vectorizer to convert the words into numerical features since Naive Bayes requires that there be a numerical input. From there, it was fairly easy to run our data through the model to train it, and then test it to make sure that it had been trained correctly.

4 Model Evaluation

Using the testing data for each of these two models, we evaluated their performance using precision, recall, and f1-score metrics.

For our bi-directional LSTM model, we ended up with a weighted average for precision of 0.28, for recall of 0.29, and for f1-score of 0.29. The full model evaluation results can be viewed in Table 1.

POS Tag	Precision	Recall	F1-score	Support
NOUN	0.36	0.39	0.37	4149
PUNCT	0.96	1.00	0.98	3113
VERB	0.26	0.28	0.27	2668
PRON	0.13	0.14	0.13	2162
PROPN	0.02	0.02	0.02	2080
ADP	0.00	0.00	0.00	2023
DET	0.01	0.01	0.01	1900
ADJ	0.14	0.12	0.13	1700
AUX	0.22	0.22	0.22	1498
ADV	0.10	0.13	0.11	1231
CCONJ	0.74	0.73	0.74	747
PART	0.41	0.35	0.38	632
NUM	0.06	0.05	0.06	536
SCONJ	0.04	0.05	0.05	388
X	0.04	0.01	0.02	139
INTJ	0.35	0.28	0.31	120
SYM	0.00	0.00	0.00	92
Accuracy			0.29	25178
Macro Avg	0.23	0.22	0.22	25178
Weighted Avg	0.28	0.29	0.29	25178

Table 1: Bi-directional LSTM Model Evaluation Scores.

For our Naive Bayes model, we ended up with a weighted average for precision of 0.70, for recall of 0.65, and for f1-score of 0.63. The full model evaluation results can be viewed in Table 2.

POS Tag	Precision	Recall	F1-score	Support
ADJ	0.87	0.79	0.83	1693
ADP	0.74	0.99	0.85	2018
ADV	0.94	0.75	0.83	1225
AUX	0.89	0.86	0.88	1495
CCONJ	1.00	0.96	0.98	739
DET	0.94	0.72	0.81	1896
INTJ	1.00	0.35	0.52	120
NOUN	0.37	0.89	0.53	4132
NUM	0.89	0.61	0.72	536
PART	0.82	0.15	0.26	630
PRON	0.91	0.75	0.82	2158
PROPN	0.92	0.36	0.52	2076
PUNCT	0.00	0.00	0.00	3106
SCONJ	0.96	0.31	0.47	387
SYM	1.00	0.00	0.00	92
VERB	0.89	0.77	0.83	2655
X	1.00	0.53	0.69	139
Accuracy			0.65	25097
Macro Avg	0.83	0.58	0.62	25097
Weighted Avg	0.70	0.65	0.63	25097

Table 2: Naive Bayes Model Evaluation Scores.

In comparing the evaluations of these two models, it is clear that the Naive Bayes model is much more accurate and precise. In terms of both macro precision and recall, as well as weighted precision and recall, we get better results with Naive Bayes.

Considering the performance of the bi-directional model, it is clear that it produced poor predictions, and it is worth noting that this type of model may not be optimal for the POS tagging problem.

However, with the very clear difference in performance between these two models, we can easily pick Naive Bayes as the better of the two to proceed with testing new data.

5 Using the Classifier on New Data

After successfully training and identifying the best model for our supervised problem, we are able to use it to annotate new corpora. The corpus that I chose to work with for this contained a large amount of sentences from newspapers. There was a huge amount of data in this corpus, so I chose to just focus on newspaper data from 2005.

After pre-processing this data, and then running it through our model, I found that the results were fairly accurate. The precision of the model's predictions matched the model's previous evaluation.

There were certainly mistakes or data that were not quite accurate, but to improve this I would suggest using more data to train the classifiers or using a different one altogether.

Even if it is not 100 percent accurate, this model appears to successfully predict POS labels for any corpora that it is used on, and could be used in the future to assist in language and sentence analysis procedures.

6 Conclusion

Using Naive Bayes and bidirectional LSTM we were able to train these two models to predict part of speech tags in corpora. Although we did see poor results from one of these models, the Naive Bayes model was demonstratively effective in this classification task.

Being able to use an effective POS prediction model like this could help out in various NLP tasks and experiments. If we are easily able to identify parts of speech, we can create more analysis about context, grammar, meaning, and other insights that you can gain from studying corpora. In the future it could be beneficial to use this model in those kinds of experiments and studies.

References

Ffatty. 2023. English news sentence corpus. https://www.kaggle.com/datasets/ffatty/english-news-sentence-corpus.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.