DOI: 10.2478/ijasitels-2024-0004

AUTOMATIC PART-OF-SPEECH TAGGING

Adelina Manolea[1],

[1]*master student in the Embedded Systems program, Faculty of Engineering, "Lucian Blaga" University of Sibiu, Romania*

adelina.manolea@ulbsibiu.ro

Abstract

Natural language processing (NLP) is a key technique in Business Process Management (BPM). The performance of BPM methods, which are based on NLP, is limited by the accuracy of automatic part-of-speech tagging, a base subtask of NLP.[9] The automatic part-of-speech tagging is the process of assigning a tag to every word in a text or a document.[1] I have developed and presented in this paper an application that learns to correctly predict parts-of-speech for words within a sentence using a machine learning algorithm. For this I used a pre-labeled data set (Brown Corpus) and implemented, evaluated and compared several versions of the n-Gram algorithm with the aim of obtaining the best classification accuracy of the automatic part-of-speech tagging process.

Keywords: part-of-speech tagging, n-Gram language model, text normalization

1 Introduction

Natural language processing is a technique that allows computers to understand human language. A correctly done part-of-speech tagging of a word supplies linguistic signals about how it is used in a sentence, and therefore it is useful for distinguishing the meaning of a word. Often words are lexically ambiguous, meaning they can have several parts of speech and depending on them several meanings. Automatic part-of-speech tagging is a disambiguation problem, its purpose focusing on ambiguous words and their correct tagging in different contexts.[1] In this paper I will present how I decided to implement, evaluate and compare several versions of the n-Gram language model with the aim of obtaining the best possible classification accuracy of the process of automatic part-of-speech tagging.

2 Application Architecture

I started developing this application with the pre-labeled Brown corpus data set and divided it into a training data set and a test data set to later evaluate the performance of the algorithms on new data. Then I preprocessed the training data set and saved the parts of speech with which each word appears and the frequency with which they appear. On the test data set I evaluated the implemented predictors, namely: the non-adaptive predictor, the 1-Gram predictor, the 2-Gram predictor and the 3-Gram predictor.

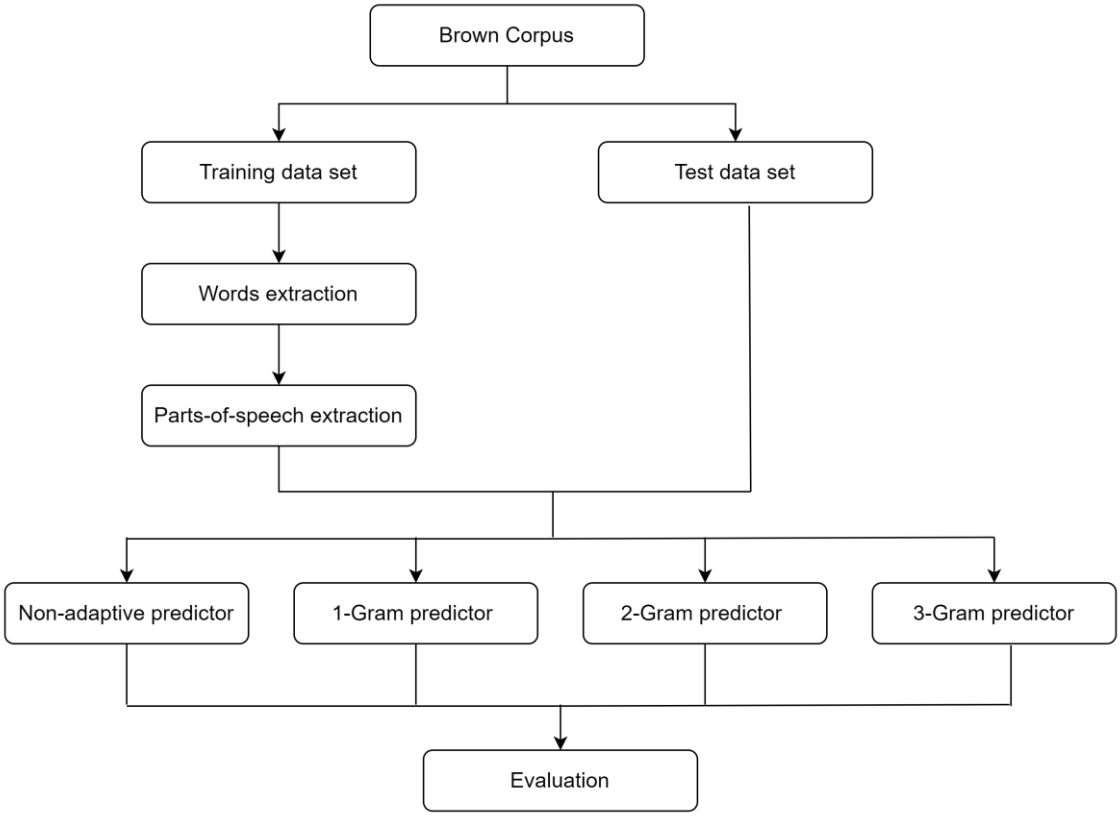

Figure 1. Application Architecture

2.1 Brown Corpus Processing

As input data for the application, I chose to use the C form of the Brown corpus, which is the grammatically tagged version. It consists of 500 files of approximately 2000 words each from 15 different domains. Each word is provided with a label that assigns it to a specific word class.
The words in the Brown corpus are of the "word/part of speech" form. I separated them by "/" and saved in a dictionary all the words, the parts of speech they appeared with and the frequency with which they appeared with those parts of speech.

Example:

Table 1. Sentence from Brown Corpus file

	0
text	The/at Fulton/np-tl County/nn-tl Grand/jj-tl Jury/nn-tl said/vbd

Table 2. Vector of words

	0	1	2	3	4	5
words	The/at	Fulton/np-tl	County/nn-tl	Grand/jj-tl	Jury/nn-tl	said/vb

Table 3. Words dictionary with the parts-of-speech and the frequency with which they were met with those parts of speech

	key	value	
		key	value
wordsList[0]	The	at	1
wordsList[1]	Fulton	np-tl	1
wordsList[2]	County	nn-tl	1
wordsList[3]	Grand	jj-tl	1
wordsList[4]	Jury	nn-tl	1
wordsList[5]	said	vbd	1

After this processing I obtained 64,735 individual words that occur with several parts of speech more than once.
Based on this dictionary, I also created a dictionary for the parts of speech with which to count the frequency with which each appears.
Example:

Table 4. Part-of-speech dictionary

	key	value
PoS[0]	at	1
PoS[1]	np-tl	1
PoS[2]	nn-tl	1
PoS[3]	jj-tl	1
PoS[4]	nn-tl	1
PoS[5]	vbd	1

I obtained 472 parts-of-speech.
To ease learning, I have reduced these parts-of-speech to 11 general parts-of-speech, namely: noun, verb, preposition, pronoun, article, adjective, conjunction, adverb, numeral, interjection and other.

After reducing the parts of speech, I obtained the following statistics:

Table 5. Parts-of-speech statistics on the entire data set after generalization

Curt. No.	PoS	Frequency of occurrence	Percentage of total words
1	noun	274336	27.30%
2	verb	197743	19.68%
3	preposition	122473	12.19%
4	pronoun	107717	10.72%
5	article	99077	9.86%
6	adjective	80741	8.03%
7	conjunction	60306	6.00%
8	adverb	48488	4.82%
9	numeral	7428	0.74%
10	other	6014	0.60%
11	interjection	627	0.06%

As expected, the noun is the most frequent part of speech with a frequency of 27.30%, followed by the verb with a frequency of 19.68%.

2.2 Splitting the data set

It is important to divide the data set into a training data set and a test data set to evaluate algorithms with new data that are part of the same domains. I chose to divide the data set into 70% training data set and 30% test data set. I did the same processing that I did on the whole data set on the training data set, and I obtained the following statistics:

Table 6. Part-of-speech statistics on the training dataset

Curt. No.	PoS	Frequency of occurrence	Percentage of training data set words
1	noun	182077	27.22%
2	verb	131981	19.73%
3	preposition	81632	12.20%
4	pronoun	71867	10.74%
5	article	65732	9.83%
6	adjective	53549	8.01%
7	conjunction	40102	5.99%
8	adverb	32635	4.88%
9	numeral	4922	0.74%
10	other	4000	0.60%
11	interjection	438	0.07%

The proportions of the general parts-of-speech are preserved.

2.3 Non-adaptive predictor

The non-adaptive predictor is the predictor that returns the most frequent part of speech every time. After evaluating the predictor on the test data set, it managed to predict 92,259 words correctly and 243,756 incorrectly, so it had an accuracy of 27.46%.

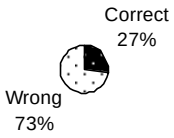

Figure 2. Non-adaptive predictor accuracy

2.4 1-Gram adaptive predictor

The 1-Gram adaptive predictor is the predictor that, if it finds the word, returns the most frequent part of speech with which it was met, and otherwise returns the most frequently met part of speech in general. After evaluating the predictor on the test data set, it managed to correctly predict 140,309 words and 195,706 incorrectly, so it had an accuracy of 41.76%.

Figure 3. 1-Gram predictor accuracy

2.5 2-Gram adaptive predictor

The 2-Gram adaptive predictor is the predictor that returns the part of speech that occurs most frequently after the part of speech of the previous word. I have implemented 3 variants of this predictor:

- Implementation based on 1-Gram predictions. The part-of-speech prediction of the current word is done with the 1-Gram algorithm's prediction of the previous word. After evaluating the predictor on the test data set, it managed to correctly predict 57,802 words and 235,330 incorrectly, so it had an accuracy of 19.72%.
- Implementation based on his own predictions. The part-of-speech of the first word in a sentence is predicted with the 1-Gram algorithm, and the rest will be predicted based on the output of the 2-Gram predictor for the previous word. After evaluating the predictor on the test data set, it managed to correctly predict 49,786 words and 243,346 incorrectly, so it had an accuracy of 16.98%.
- Largest limit available. This implementation of the adaptive 2-Gram predictor aims to achieve the highest possible performance by starting from the correct part of speech. This assumes that instead of using the 1-Gram predictor, the correct part-of-speech of sentence-beginning words will be read directly from the test data set. This will eliminate the highly likely possibility of mislabeling an entire sentence just because the 1-Gram predictor mispredicted the first word in the sentence. After evaluating the predictor on the test data set, it managed to correctly predict 110,784 words and 225,231 incorrectly, so it had an accuracy of 32.97%

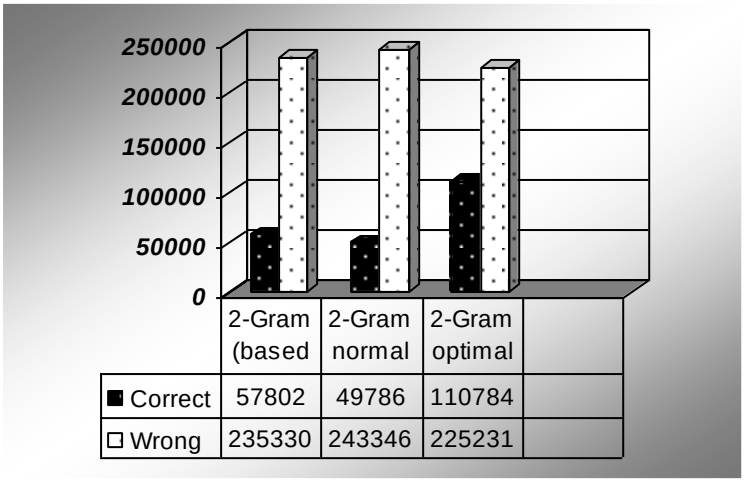

Figure 4. 2-Gram predictors accuracy

2.6 3-Gram adaptive predictor

The 3-Gram adaptive predictor is the predictor that returns a part of speech considering both the part of speech of the previous word and the part-of-speech of the posterior word. Thus, the predictor will evaluate sequences of 3 words and return the most frequent 3-word sequence that has as neighboring parts of speech the parts of speech of the preceding word and the following word. After evaluating the predictor on the test data set, it managed to correctly predict 110,784 words and 199,778 incorrectly, so it had an accuracy of 31.81%

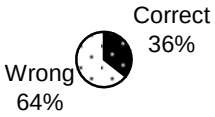

Figure 5. 3-Gram predictor accuracy

3 Experimental Results

After training the presented algorithms on the training data set, I obtained the following evaluation performance on the test data set:

Curt. No.	Algorithm	No. of correct predictions	No. of wrong predictions	Accuracy
1	Non-adaptive	92259	243756	27,46%
2	1-Gram	140309	195706	41,76%
3	2-Gram (based on 1-Gram)	57802	235330	19,72%
5	2-Gram normal	49786	243346	16,98%
6	2-Gram optimal	110784	225231	32,97%
7	3-Gram	93207	199778	31,81%

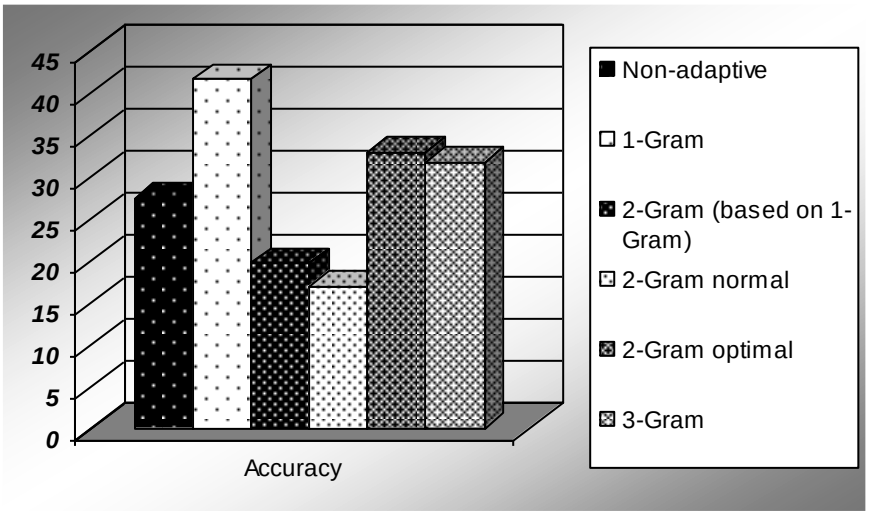

Figure 6. Accuracy of presented algorithms

4 Conclusions

The presented part-of-speech automatic tagging algorithm was based on the different forms of the n-Gram language model. Following the evaluations, I obtained the best performance in terms of classification accuracy using the 1-Gram adaptive algorithm. So, the maximum accuracy I was able to achieve was 41%.

Although I tried to get better performance by increasing the degree of the n-Gram algorithm, by evaluating more neighboring words to widen the context, I could not achieve better performance than the 1-Gram algorithm.

The n-Gram algorithm is not the best performing machine learning algorithm for part-of-speech automatic tagging, but it is a basic tool for understanding the fundamental concepts of language modeling.

References

[1] Dan Jurafsky, James H. Martin, *Speech and Language Processing*, available online at https://web.stanford.edu/~jurafsky/slp3/
[2] https://devopedia.org/n-gram-model , accessed in 05.2022
[3] https://www.ibm.com/cloud/learn/text-mining , accessed in 05.2022
[4] http://korpus.uib.no/icame/manuals/BROWN/INDEX.HTM#bc1 – Brown Corpus Manual, accessed in 05.2022
[5] https://en.wikipedia.org/wiki/Part_of_speech, accessed in 05.2022
[6] https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234 , accessed in 06.2022
[7] https://machinelearningmastery.com/how-to-evaluate-machine-learning-algorithms/ , accessed in 06.2022
[8] https://www.jeremyjordan.me/evaluating-a-machine-learning-model/ , accessed in 06.2022
[9] https://research.ibm.com/publications/a-novel-part-of-speech-tagging-framework-for-nlp-based-business-process-management?mhsrc=ibmsearch_a&mhq=part%20of%20speech%20tagging , accessed in 06.2022