Part-of-Speech Tagging

Part-of-speech tagging is the process of assigning a part-of-speech to each word in a text.

General requirements:
Create an application that learns to automatically label Parts of Speech for each word in a sentence that was read from the keyboard or from an input file. To train the algorithm, use Brown Corpus dataset.

1 Brown Corpus Processing

The Brown Corpus represents a general collection of texts that is used in natural language processing research and has been manually created by professors Henry Kucera and W. Nelson Francs. This dataset contains several documents where each word contains a label with its part of speech. The Brown corpus contains 1014312 words that were structured in 500 documents. Each document contains entire sentences and 2000 or more than 2000 words.

1.1 Requirements:

Make a module (application) that opens all files from the Brown Corpus and extracts all the words from the file and their corresponding part of speech (PoS). It is recommended to create a structure in which for each word are kept all PoS with the word appeared in the corpus. The number of occurrences of the word with the respective PoS will be counted.
At the end, some statistical information needs to be generated like:
- the total number of distinct words in the corpus
- the total number of distinct parts of speech in the corpus
- the number of occurrences for each PoS separately
- ...

1.2 General aspects

In the tagged version of Brown Corpus each individual word is furnished with a brief tag which assigns it to a specific word-class. There are 82 of these tags that are of six kinds:

1. mayor form-classes («parts of speech»): noun, common and proper; verb; adjective; adverb; in short, the open lexical classes;
2. function words: determiners, prepositions, conjunctions, pronouns, etc.; the closed lexical and grammatical classes;
3. certain important individual words: *not*, existential *there,* infinitival *to,* the forms of the verb, do, *be,* and *have* , whether auxiliaries or full verbs;
4. punctuation marks of syntactic significance;
5. inflectional morphemes, notably noun plural and possessive; verb past, present and past participle, and 3rd singular concord marker; comparative and superlative adjective and

adverb suffixes. Thus, when the following symbols appear elsewhere than in the first two letters of a tag, they normally have the following equivalences:

S = plural
D = past tense
$ = possessive
Z = 3rd singular verb
R = comparative
N = past participle
T = superlative
G = present participle or gerund
O = objective case of pronoun

6. two tags, FM and NC, are hyphenated to the regular tags to indicate that a word is a foreign word or a cited word, respectively. Thus the word *de* tagged FW-IN indicates that it is a foreign preposition. In a sentence such as «The word *if* has two letters,» *if* would be tagged CS-NC. (The tag -HL is hyphenated to the regular tags of words in headlines. The tag - TL is hyphenated to the regular tags of words in titles).

1.3 List of tags

No.	Tag	Description	Examples
1.	.	sentence closer	. ; ? !
2.	(	left paren	
3.	)	right paren	
4.	*	not, n't	
5.	--	dash	
6.	,	comma	
7.	:	colon	
8.	ABL	pre-qualifier	quite, rather
9.	ABN	pre-quantifier	half, all
10.	ABX	pre-quantifier	both
11.	AP	post-determiner	many, several, next
12.	AT	article	a, the, no
13.	BE	be	
14.	BED	were	
15.	BEDZ	was	
16.	BEG	being	
17.	BEM	am	
18.	BEN	been	
19.	BER	are, art	
20.	BEZ	is	
21.	CC	coordinating conjunction	and, or
22.	CD	cardinal numeral	one, two, 2, etc.

23.	CS	subordinating conjunction	if, although
24.	DO	do	
25.	DOD	did	
26.	DOZ	does	
27.	DT	singular determiner	this, that
28.	DTI	singular or plural determiner/quantifier	some, any
29.	DTS	plural determiner	these, those
30.	DTX	determiner/double conjunction	either
31.	EX	existential *there*	
32.	HV	have	
33.	HVD	*had* (past tense)	
34.	HVG	having	
35.	HVN	*had* (past participle)	
36.	HVZ	has	
37.	IN	preposition	
38.	JJ	adjective	
39.	JJA	adjective + Auxiliary	
40.	JJC	adjective, Comparative	
41.	JJCC	Adjective + Conjunction	
42.	JJR	comparative adjective	
43.	JJS	semantically superlative adjective	chief, top
44.	JJF	Adjective + Female	
45.	JJT	morphologically superlative adjective	biggest
46.	JJM	Adjective + Male	
47.	MD	modal auxiliary	can, should, will
48.	NN	singular or mass noun	
49.	NNA	Noun + Auxiliary	
50.	NNC	Noun + Conjunction	
51.	NN$	possessive singular noun	
52.	NNS	plural noun	
53.	NNS$	possessive plural noun	
54.	NP	proper noun or part of name phrase	
55.	NP$	possessive proper noun	
56.	NPS	plural proper noun	
57.	NPS$	possessive plural proper noun	
58.	NR	adverbial noun	home, today, west

59.	NRS	plural adverbial noun	
60.	NNP	proper noun or part of name phrase	
61.	NNPC	proper noun + Conjunction	
62.	OD	ordinal numeral	first, 2nd
63.	PN	nominal pronoun	everybody, nothing
64.	PN$	possessive nominal pronoun	
65.	PP$	possessive personal pronoun	my, our
66.	PP$$	second (nominal) possessive pronoun	mine, ours
67.	PPL	singular reflexive/intensive personal pronoun	myself
68.	PPLS	plural reflexive/intensive personal pronoun	ourselves
69.	PPO	objective personal pronoun	me, him, it, them
70.	PPS	3rd. singular nominative pronoun	he, she, it, one
71.	PPSS	other nominative personal pronoun	I, we, they, you
72.	PRP	personal pronoun, singular	
73.	PRPS	personal pronoun, plural	
74.	PRP$	Possessive pronoun	
75.	PUN	All Punctuations	
76.	SYM	Symbols	
77.	QL	qualifier	very, fairly
78.	QLP	post-qualifier	enough, indeed
79.	RB	adverb	
80.	RBR	comparative adverb	
81.	RBS	superlative adverb	
82.	RBT	superlative adverb	
83.	RN	nominal adverb	here then, indoors
84.	RP	adverb/particle	about, off, up
85.	TO	infinitive marker *to*	
86.	UH	interjection, exclamation	
87.	VB	verb, base form	
88.	VBA	verb + Auxiliary, singular, present	
89.	VBD	verb, past tense	
90.	VBG	verb, present participle/gerund	
91.	VBN	verb, past participle	
92.	VBZ	verb, 3rd. singular present	
93.	WDT	*wh*- determiner	what, which
94.	WP$	possessive *wh*- pronoun	whose

95.	WPO	objective *wh-* pronoun	whom, which, that
96.	WPS	nominative *wh-* pronoun	who, which, that
97.	WQL	*wh-* qualifier	how
98.	WRB	*wh-* adverb	how, where, when

Some important observations:

1. Not all tuples of form <word/tag> contain only one symbol "/". There are tuples that contain 2 or 3 "/" symbols, for example "1-1/2/cd" or "and/or/cc". In this case the PoS is on the last position and can be ease extracted. The problem is with the word, in some case the word cand be kept with symbol "/" as in the first case from example but in the second case recommend extracting 2 different words noted with same PoS.
2. Merged constructions are marked by the appropriate tags joined by +, except for *, which is affixed directly.
    a. For example: for the tags that contain symbol '+' we can keep only the values before the symbol '+'.
    b. The tags FW, NC, NP are given to single words, phrases, or sentences contained respectively in foreign phrases, cited word, and compound or complex titles or names. For the tags that start with "FW-" we can keep the value after "FW-"
    c. The tag -HL is hyphenated to the regular tags of words in headlines, recommend eliminating this part from tag.
    d. The tag -TL is hyphenated to the regular tags of words in titles, recommend eliminating this part form tag.
3. "Nil" - in the data set there are words/sentences where the part of speech is not specified, in this case the "nil" tag is used to specify that word not have a PoS.

More detailed information are at http://korpus.uib.no/icame/manuals/BROWN/INDEX.HTM .

1.4 Reducing the number of parts of speech

Since it is more difficult to create an application that learns all the parts of speech found in the Brown Corpus dataset (how many did you find?), I recommend reducing their number to a maximum of 10 basic parts of speech (noted #PoS). I recommend that the remaining parts be the basic ones and be represented in a balanced way in the data set. No part of speech will be deleted, only certain parts of speech will be grouped under the same name.
For example:

Tag	All PoS name	New Pos Name
BE	be	VB
BED	were	
BEDZ	was	
BEG	being	
BEM	am	
BEN	been	
BER	are, art	

BEZ	is	
VB	verb, base form	
VBA	verb + Auxiliary, singular, present	
VBD	verb, past tense	
VBG	verb, present participle/gerund	
VBN	verb, past participle	
VBZ	verb, 3rd. singular present	

With the parts of speech that do not fit into any category created above, a distinct category will be created called, for example, Other.
I recommend the following classes to be created: Nouns, verbs, adjectives, adverbs, interjections, others, ….

1.5 Some rules that can be applied directly:

In English language there are some rules that can be applied in the PoS process, for selecting the best PoS:

- nouns in English are preceded by determiners and adjectives;
- verbs are preceded by nouns;
- articles (a, an, the) often precede nouns;
- there is a syntactic structure (verbs have dependency links to nouns);
- words "a," "an," and "the," are usually articles and indicate that a noun is coming;
- words like "and," "but," and "or" are typically conjunctions;
- is a word end with:
    - -ly - Often indicate an adverb
    - -er – Often indicate an adjective
    - -est – Often indicate an ajective
    - -ing – Often indicate a verb
    - -ed – Often indicate a verb
    -
- ….

If there are some fix rules it is better to be applied before apply a learning algorithm.

1.6 Reducing the number of words form the corpus.

- Eliminating the stop words
- Extraction of the stem of the word
- Extraction of the lemma of the word
- …

2 Methods for tagging words

2.1 Non-Context-Awareness Tagging Methods

2.1.1 Most frequently PoS -MFPoS_1

The first proposed method predicts all the time the most frequently part of speech that occur in the training dataset.

2.1.2 Basic Global Probability – BGP

This method uses only the Brown Corpus database for predicting part of speech. Thus, using the training set we have counted separately for each word all parts of speech which appear for that word in the training data set. The BGP method computes the probability of a word to be a noun, a verb, an adverb, an adjective, or another part of speech using the training set. These probabilities are obtained in the training phase and represent the ratio between the total number of occurrences of the given word when it was labeled with a certain tag (noun, verb, adverb, etc.) and the count of all occurrences of the given word in the training set. We note this type of probability as P_{GP}. Thus, for each word that appears in the training set we must compute for each part of speech in which the word appears the probability, so that at the end of training we can compute the most likely type of part of speech for that word.
Choosing the tag for a word in the test phase is made using the following formula (we choose the most probable type of part of speech for the word computed in the training phase):

$$\text{Label} = \underset{pos \in \vartheta}{\operatorname{argmax}} \left(P_{GP_{pos}} \right), \quad where \ \vartheta = \{verb, noun, adv, adj, \dots\} \tag{2.1}$$

The results of this method depend on how complex and distributed the training set is. Unfortunately, this method will not be able to predict several types of speech for the same word. Also, there is still a problem when a word appears in the test dataset, and it was not present in the training dataset. In this case the word will be tagged with the label "not_found".

2.1.3 Basic Global Probability_2 – BGP_2

This method is the same as the previous method with 2 changes:

1. If a word contains only numbers, then we predict it as a Numeral, if have this PoS (without looking for it in the training dataset because it is unlikely to find the exact same number).
2. If we don't find that word in the training data set, we predict it as having the part of speech "Noun".

2.2 Statistical Context-Awareness Tagging Methods

The methods presented in this section attempt to predict the part of speech for a word based on the context in which this word appears in the sentence. To include the context in which the word appears in the training set, we need to compute some probabilities for training the Naive Bayes classifier. The Naive Bayes classifier in this context does not computes the likelihood of appearance of a certain part of speech based on a certain class. Here it computes the probability of occurrence for the part of speech of a word conditioned by the part of speech of the previous word, next word or both words.

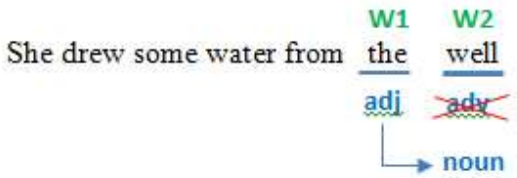
Can be used the following general formula for the Naïve Bayes classifier.

$$pos_{NB} = \underset{1<i<\#\text{PoS}}{\operatorname{argmax}} P(Y_i) * \prod_{j=0}^{\#PoS} P(x_j|Y_i) \quad (2.2)$$

The index *i* represents the part of speech for the word that need to be tagged. The index *j* represents the part of speech of the previous or of the next word (depending on the used method). The likelihood $P(Y_i)$ is a value that indicates the relative frequency for the label Y_i for the current word. The likelihood $P(x_j|Y_i)$ from the previous equation computes how "suitable" the label Y_i (part of speech) is for current word if that the neighbour word has the tag x_j. The sum of these two probabilities is a value that indicates the part of speech for a particular word.
The equation computes the most significant label for a specific word. Because of the logarithm the obtained values will be negative numbers.

2.2.1 Backward Naïve Bayes - BNB

This method computes the part of speech based on Global Probability taken from the training set (as in section above) and the part of speech from the previous word.
Let's take the following example:



In this example, if we take the word "well" and individual analyse it based only on the information retrieved from training set we will find that it is most used as an adverb, so for this word there is a small probability to choose as label another part of speech. From this point of view, we have tried to influence the corresponding probability of a label associated to a particular word using the computed Naïve Bayes classifier probabilities.
After applying the BNB method, the word "well" will be labelled as a noun because of the article "the" to the left side of it.
To include in the prediction this aspect we have computed in the training step a matrix where we have stored the number of intersections of the two words W1 and W2 which will be used by the Backward Naïve Bayes method in the testing step. Thus, at the end of the training step we have for each word a vector that contains the total number of occurrences of the particular word and its number of occurrences under different labels. The resulting matrix after training is a square matrix which has the dimension equal to *the number of words in the training set* * *no. the parts of speech* and stores the number of intersections between any two words *W1* and *W2* for each label.
In the test phase we use the Backward Naïve Bayes method to predict the part of speech for the words extracted from the test set, based on the following information: probability of the occurrence of certain parts of speech related to the current word and the conditional probability between the occurrence of words in some order and their appropriate types of the part of speech.

$$pos_{BNB} = \underset{i \in \#PoS}{\operatorname{argmax}}\left[P(W2_i) * \prod_{j \in \vartheta} P\left(W1_j|W2_i\right)\right] \quad (2.3)$$

In the next equation where *RB* means adverb and *VB* means verb we exemplify how we compute the conditional probability from the equation 2.3. It expresses the probability that the word *W1* is an adverb conditioned that *W2* is a verb.

$$P(W1_{RB} \mid W2_{VB}) = \frac{\#\_(W1_{RB} \cap W2_{VB})}{\#\_(W2_{VB})} \tag{3.6}$$

where $\#\_(W1_{RB} \cap W2_{VB})$ indicates how many times the two "words" $W1_{RB}$ and $W2_{VB}$ intersected (i.e. how many times the word *W1* appeared as adverb followed by the word *W2* as a verb) and $\#\_(W2_{VB})$ represents the number of occurrences of the word *W2* as a verb.
For example, for two words *W1* and *W2* the intersection matrix would look like (Table 3.1):

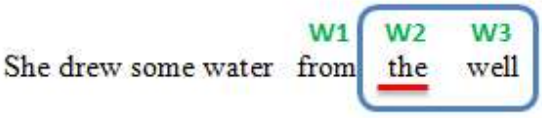
W1 ↓ / W2 →	VB	NN	RB	JJ	…	OTH
VB	$W1_{VB} \cap W2_{VB}$	$W1_{VB} \cap W2_{NN}$	$W1_{VB} \cap W2_{RB}$	$W1_{VB} \cap W2_{JJ}$		$W1_{VB} \cap W2_{OTH}$
NN	$W1_{NN} \cap W2_{VB}$	$W1_{NN} \cap W2_{NN}$	$W1_{NN} \cap W2_{RB}$	$W1_{NN} \cap W2_{JJ}$		$W1_{NN} \cap W2_{OTH}$
RB	$W1_{RB} \cap W2_{VB}$	$W1_{RB} \cap W2_{NN}$	$W1_{RB} \cap W2_{RB}$	$W1_{RB} \cap W2_{JJ}$		$W1_{RB} \cap W2_{OTH}$
JJ	$W1_{JJ} \cap W2_{VB}$	$W1_{JJ} \cap W2_{NN}$	$W1_{JJ} \cap W2_{RB}$	$W1_{JJ} \cap W2_{JJ}$		$W1_{JJ} \cap W2_{OTH}$
…						
OTH	$W1_{OTH} \cap W2_{VB}$	$W1_{OTH} \cap W2_{NN}$	$W1_{OTH} \cap W2_{RB}$	$W1_{OTH} \cap W2_{JJ}$		$W1_{OTH} \cap W2_{OTH}$

Table 3.1: The intersection matrix for BNB

The disadvantage of this method is that it will predict the part of speech of a word based on the part of speech predicted for the previous word which can lead to an error propagation in case that at a moment it has predicted a wrong type of speech for a specific word. Nevertheless, this method has the advantage that it is based on the type of speech of the previous word, fact that happens also in the natural language.

2.2.2 Forward Naïve Bayes - FNB

The FNB method will predict the part of speech of a word based on the global probability (BGP - the most significant part of speech for the given word) extracted from the training set and the part of speech for the next word from the sentence.
For example, if we consider the next sentence, we try to predict the part of speech of the word *W2* based on the part of speech of the word *W3* (the part of speech for the word *W3* must be also predicted).



To embed this prediction in the training stage we used a matrix (Table 3.2) which stores the number of intersections between the two words *W2* and *W3* (example ($W3_{RB} \cap W2_{VB}$)) for be used by the Forward Naïve Bayes method.

W3 ↓ / W2 →	VB	NN	RB	JJ	…	OTH
VB	$W3_{VB} \cap W2_{VB}$	$W3_{VB} \cap W2_{NN}$	$W3_{VB} \cap W2_{RB}$	$W3_{VB} \cap W2_{JJ}$		$W3_{VB} \cap W2_{OTH}$
NN	$W3_{NN} \cap W2_{VB}$	$W3_{NN} \cap W2_{NN}$	$W3_{NN} \cap W2_{RB}$	$W3_{NN} \cap W2_{JJ}$		$W3_{NN} \cap W2_{OTH}$
RB	$W3_{RB} \cap W2_{VB}$	$W3_{RB} \cap W2_{NN}$	$W3_{RB} \cap W2_{RB}$	$W3_{RB} \cap W2_{JJ}$		$W3_{RB} \cap W2_{OTH}$
JJ	$W3_{JJ} \cap W2_{VB}$	$W3_{JJ} \cap W2_{NN}$	$W3_{JJ} \cap W2_{RB}$	$W3_{JJ} \cap W2_{JJ}$		$W3_{JJ} \cap W2_{OTH}$
…						
OTH	$W3_{OTH} \cap W2_{VB}$	$W3_{OTH} \cap W2_{NN}$	$W3_{OTH} \cap W2_{RB}$	$W3_{OTH} \cap W2_{JJ}$		$W3_{OTH} \cap W2_{OTH}$

Table 3.2 – The intersection matrix for FNB

Thus, at the end of the training phase we have for each word a vector which stores the total number of occurrences of that word and the occurrences for each part of speech for that word. Also, a matrix is stored with the number of intersections between the two words *W3* and *W2* for each part of speech.
In the test phase we use the Forward Naïve Bayes method to predict the part of speech for the words extracted from the test set using the following formula:

$$pos_{FNB} = \underset{i \in \#PoS}{\operatorname{argmax}}\left[P(W2_i) * \prod_{j \in \vartheta} P\left(W2_j \middle| W1_i\right)\right] \quad (3.7)$$

The computed probabilities are similar with those computed in the equation 3.5.

2.2.3 Complete Naïve Bayes - CNB

You can combine the last two methods presented above so that they predict each method separately and finally choose the best result. A confidence factor can be introduced to increase the chance of correctly predicting the PoS, depending on the context of the word to be predicted.
To predict the part of speech for the first word in the sentence we use only the FNB method, and for the last word in the sentence we use only the BNB method.

3 Evaluation metrics

Measuring the performance of tagging methods is performed using and combining different evaluation metrics. These metrics indicate how good or correct is the method to predict a label for a particular word. We propose to use five different measures: Accuracy, Precision, Recall, F-measure and True Negative Rate.
Precision – represents the percentage of words correctly tagged into a category over those that have been tagged in that category.

$$Precision = \frac{\#\_retrieved\_relevant\_tags}{\#\_retrieved\_tags} \quad (3.1)$$

Recall - represents the percentage of relevant words correctly labeled into a category over all relevant words from that category

$$Recall = \frac{\#\_relevant\_retrieved\_tags}{\#\_relevant\_tags} \quad (3.2)$$

Precision and recall ranges from 1 to 0. In fact *precision* represents a quantitative measure of the information retrieval system while *recall* represents a qualitative measure of this system.
F-measure – represents the harmonic mean between Precision and Recall. This measure penalizes better comparatively with other measures (as arithmetic mean) a system that is influencing more a metric over the other.

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \quad (3.3)$$

For compute all this evaluation metrics we recommend computing for each PoS a confusion matrix (contingency table) that is 2x2 matrix for each PoS.

Confusion matrix for a specific PoS;.

PoS		Predicted class	
		Class=YES	Class=NO
Actual Class	Class=YES	*True Pozitive (tp)*	*False Negative (fn)*
	Class=NO	*False positive (fp)*	*True Negative (tn)*

Accuracy:

$$Accuracy = \frac{tp+tn}{tp+fn+fp+tn} \quad (3.4)$$

Precision or positive predictive value:

$$\text{Precision} = \frac{tp}{tp+fp} \quad (3.5)$$

Recall, sensitivity or true positive rate:

$$\text{Recall} = \frac{tp}{tp+fn} \quad (3.6)$$

Specificity, selectivity or true negative rate:

$$True\ Negative\ rate = \frac{tn}{fp+tn} \quad (3.7)$$

After displaying the evaluation metrics for each individual class, the evaluation of the algorithm is the arithmetic mean for all the obtained values for that metric.
More information about evaluation metrics: https://en.wikipedia.org/wiki/Precision_and_recall