Optimizing efficiency in learning English words and expressions

By

Majdi Alali

Index:

I. Introduction:

i. Overview:

Many of us would like to learn new languages. But we get panic by thinking of the time limitation and the tons of words and expressions every language has. “Expression” can be defined as a group of words having a particular meaning used in a particular context. But do we need all these words and expressions to start **natural** conversations with native speakers? The answer is simply “no”. What we need to learn, in additional to grammar and pronunciation, is what native speakers use in everyday language. If we look closely on what native speakers say on daily basis, we will observe that it is nothing but a bunch of phrases or expressions. There are many types of phrases but in this project, we will limit ourselves to phrasal verbs. A phrasal verb consists of a verb followed by adverb(s) or preposition(s), in linguistics called particles. But the question, where can we get such an overview over the everyday language expressions? And if we find them somewhere, how can we **scientifically** know which expressions are more frequent than others in a language to learn first? Actually, there are some efforts from big institutes, like Oxford and Cambridge, to document or record the spoken language in something called speech corpora [8] but these corpora are unavailable to everyone, and we think they usually are not big enough to have an overview over everyday language because recording real conversations and updating them are time-consuming and very expensive. Anyway, doing research on huge number of speech data can be a very challenging task for CPU-es. In this project, we are going to try to find out a somehow reasonable approach to deal with the problem for English language. After lots of searching and thinking, we come up with that series’ subtitles can be a very good and cheap alternative to corpora. They can be considered as a reflection of real conversations where series' conversations are usually drawn from everyday language. In a nutshell**,** our model tries to reduce the number of vocabularies a second-language learners need to learn to **communicate** in a new language, like English in this case, as **natural** as possible. In other words, the model tries to optimize learning the shortcuts in a **spoken** language by analyzing subtitles of chosen series.

ii. Data set

Our data set is a raw txt-file, about 140000 lines, which includes subtitles from 8 diverse series [1]. Most of them are American. These series have been chosen from different genres like historic, dramatic, comic, romantic and so on to somehow avoid bias in our results. Anyway, these series are not equally big.

Obs. We could not attach the data set on Inspera because of the limitation of uploading more than to files. Anyway, you can access the data set here.

iii. NLTK model

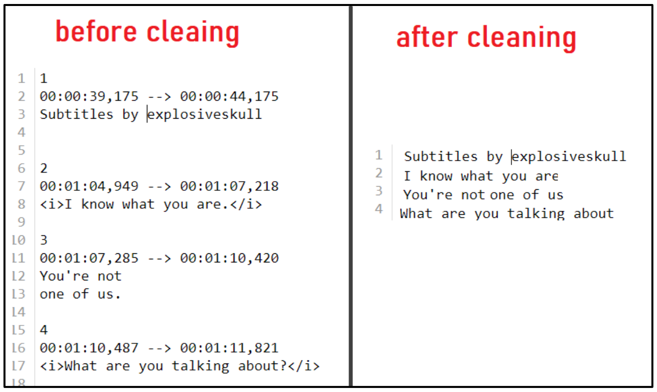
We will mainly use a model called Natural Language Toolkit. NLTK is a pre-trained statistical NLP model. [2] It has many useful corpora and tools which can deal with natural languages. Some of the common tools the model offers are tagging, lemmatizing, and stemming. Pos-tagging is the most used one. According to Wikipedia, POS is a category of words or lexical items that have similar grammatical properties like nouns, verbs and so on. We will come back to it later in the methodology section.

Obs. the attached code is totally written by me, except the nltk-tools, some basic libraries like pandas, and matplotlib and one simple method borrowed from Stack Overflow-website [7].

II. Methodology

i. Data Cleaning

It is the most important and time-consuming process. It starts with removing line breaks, timelines, and numeric representation of utterances in the series [5]. Then it gets rid of special characters like punctuation marks, by checking every single char in every single word. After that it removes some html-tags like <i> </i>. At the end it excludes function words, a list of meaningless given by nltk. But this list does not cover alle unwanted words. Therefore, we extended this list with new words and fragments. We want to pay your attention to that the data cleaning got implemented on stages and in case of need. For example, we could not delete some functional words, like prepositions, from the beginning because we would need them to extract phrasal verbs in later stages.



```
1
00:00:39,175 --> 00:00:44,175
Subtitles by explosiveskull


2
00:01:04,949 --> 00:01:07,218
<i>I know what you are.</i>

3
00:01:07,285 --> 00:01:10,420
You're not
one of us.

4
00:01:10,487 --> 00:01:11,821
<i>What are you talking about?</i>
```

```
 Subtitles by explosiveskull
 I know what you are
 You're not one of us
What are you talking about
```

Figure 1 (a data sample)

ii. Tagging and Lemmatization

a. Why we need them

As we mentioned earlier, one of our goals we want to achieve in this project is finding the most frequent English words via filtering raw texts. One typical challenge of pc while dealing with natural language is inflections of words. Inflection can be defined as the process of word formation in which items are added to the base form of a word to express grammatical functions, and to mark such distinctions as tense, person, number, gender, mood, voice, and case. For example, the computer will recognize the following words: “go”, “goes”, “went” and “going”, as four different words, not as one word if we do not tell it explicitly via implementation of tagging and lemmatization on the words.

b. Definitions

According to www.nltk.og , the process of classifying words into their parts of speech and labeling them accordingly is known as POS-tagging, or simply tagging. Parts of speech are also known as word classes or lexical categories like nouns, verbs and so on. While lemmatization is defined as a morphological analysis of words or cutting the edges which words can stand without. In linguistics, these edges called affixes. In other words, it is the process of returning the word to its base or dictionary form, which is known as the lemma of a word. For example, the lemma of ‘found’, ‘finding’, and ‘finds’ is ‘find’.

c. How they work

POS tagging is a supervised learning solution that uses features like the previous, the next word, and first letter capitalized or not etc. While the lemmmatizer, in help with Regular Expression, looks at the letters which a word starts or/and ends with, called in linguistics prefixes and suffixes, like ‘re’, ‘dis’, ‘ing’, and ‘ed’, and removes them. In addition, it does inflection on words by searching in a map of the word-family to give the contextual base form of that word.

d. **Treebank- vs. WordNet-tagger:**

Treebank-tagger is a powerful tool where it has 36 tags(=classes). This gives a precise POS of a word. While WordNet-tagger is a little poor because it has just five POS-tags: ‘n’ stands for nouns, ‘v’ for verbs, ‘j’ for adjectives, and ‘s’ and ‘a’ for adverbs. WordNet has a good lemmatizer but Treebank has not. In this project we would like to take the benefits of the tagger of Treebank- and lemmatizer of WordNet. It is important to mention that WordNet-lemmatizer only understand POS given by WorNet-tagger. Therefore we need to map between TreeBank- and WordNet-tags to be able to use Treebank-lemmatizer with combination WordNet-tagger.

iii. Statistical model (as a helping tool)

We took initiative to build from scratch a model which provides some basic functions of statistics. It will somehow substitute the use of external libraries like stats. The following functions are Among the ones we made are median, standard deviation, quartiles, z scores, and coefficient of variance. We found useful to make a function which detect outliers as well.

III. Results

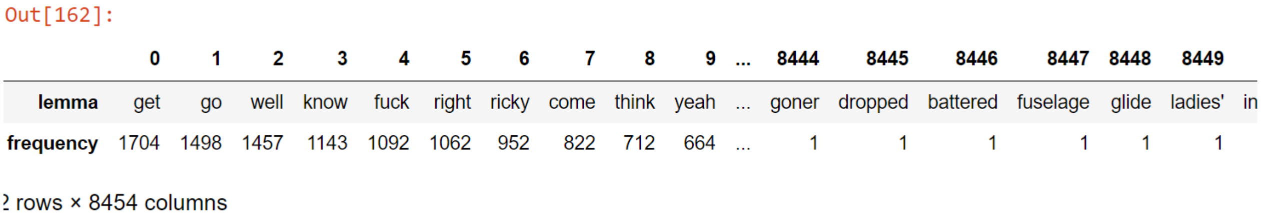
i. Overview

We have successfully achieved the main parts of our aimed goals which give optimal solutions as possible and dropped out nice-to-have parts like doing statistics on a WordNet corpus or database which include 0.5 million English words.

ii. Frequency tables:

In this section, we will show 3 frequency tables related to lemmas, phrasal verbs, and Treebank-tags. And then we will comment on every table. After that we are going to do some statistics on the first two tables.

a. lemmas



Out[162]:

	0	1	2	3	4	5	6	7	8	9	...	8444	8445	8446	8447	8448	8449	
lemma	get	go	well	know	fuck	right	ricky	come	think	yeah	...	goner	dropped	battered	fuselage	glide	ladies'	in
frequency	1704	1498	1457	1143	1092	1062	952	822	712	664	...	1	1	1	1	1	1	

2 rows × 8454 columns

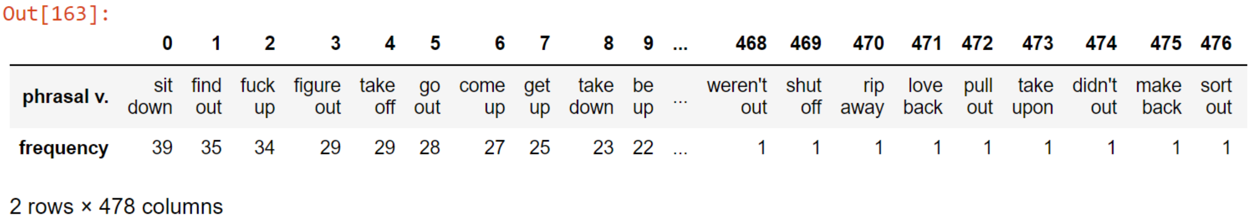
Table 1

- **comments**

- Adding few thousands of lines to data does not make a big difference in number of lemmas but It does with frequency.

- The total number of lemmas here is very low compared with the total number of English words and this is expected.

b. Phrasal verbs



Out[163]:

	0	1	2	3	4	5	6	7	8	9	...	468	469	470	471	472	473	474	475	476
phrasal v.	sit down	find out	fuck up	figure out	take off	go out	come up	get up	take down	be up	...	weren't out	shut off	rip away	love back	pull out	take upon	didn't out	make back	sort out
frequency	39	35	34	29	29	28	27	25	23	22	...	1	1	1	1	1	1	1	1	1

2 rows × 478 columns

Table 2

- **Comments**

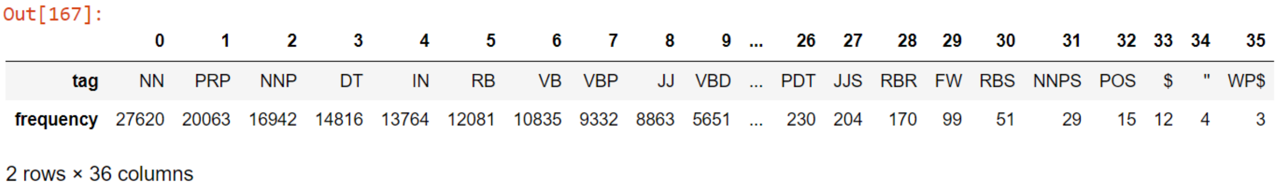
-Some of most frequent lemmas which their POS are **'verbs',** like 'come' 'get' and 'take', appear in the most frequent phrasal verbs like 'come up'. This finding is so important in further work on this project.

-The sum of phrasal verbs is very low compared with lemmas. This is not surprising because our main goal is optimizing the efficiency in learning everyday English language.

-Our phrasal verbs consists of a verb and one particle.

c. Treebank vs. WordNet tags

c1. Treebank-tags



Out[167]:

	0	1	2	3	4	5	6	7	8	9	...	26	27	28	29	30	31	32	33	34	35
tag	NN	PRP	NNP	DT	IN	RB	VB	VBP	JJ	VBD	...	PDT	JJS	RBR	FW	RBS	NNPS	POS	$	''	WP$
frequency	27620	20063	16942	14816	13764	12081	10835	9332	8863	5651	...	230	204	170	99	51	29	15	12	4	3

2 rows × 36 columns

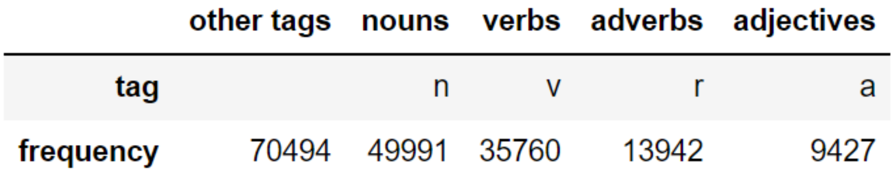
Table 3

- **comments**

-We observe that singular nouns are the most frequent tag among all others, followed by pronouns.

-It is not surprising that DT (=determiners) like ‘the’ and pronouns like ’us’ have high frequency scores because they have syntactical, usually not semantical, purposes in languages.

c2. WordNet-tags



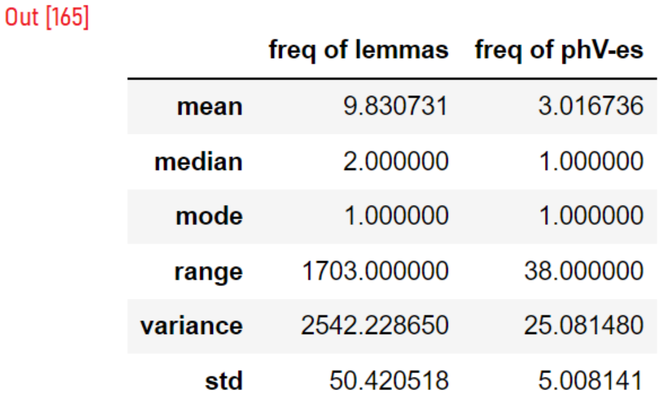
	other tags	nouns	verbs	adverbs	adjectives
tag		n	v	r	a
frequency	70494	49991	35760	13942	9427

Table 4

- **Comment**
- WordNet-tagger gives a better overview over the main tags but TreeBank-one gives more details about the syntactic characteristics of the words.

d. Statistics on the tables

The figure below shows the implementation of our statistical model on the outputs (=the prev. tables):



Out [165]

	freq of lemmas	freq of phV-es
mean	9.830731	3.016736
median	2.000000	1.000000
mode	1.000000	1.000000
range	1703.000000	38.000000
variance	2542.228650	25.081480
std	50.420518	5.008141

Table 5

- **Observations:**

-If we take the first 10 data points in *table1*. We will see immediately how wide the range is between the first data point and the tenth data point (1704-664=1040). While the case is different with the first 10 data points in *table2 (39-22=17)*. This can explain the big difference between *table1* and *table* 2 (= lemmas and phrasal verbs' frequency) regarding the range, variance, and standard deviation. Another thing which supports this claim is that the dispersion of data points in *table1* is 8400 whereas the spread of the datapoint in *table2* is just 476.

-The info. about median shows that half of lemmas, or more, appear 2 times in the series while half of the phrasal verbs, or more, appear just once!

-The mode shows that there is something in common between table1, and table 2. This thing is that the most frequent value of the frequency of phrasal verbs and lemmas is 1!

- The range do not always give a good impression of the variability of the data, because it deals with data which can be outliers. But in our case it helps to give a better understanding to the the variability of the data.

iii. Accuracy

As we mentioned earlier, our data is raw (= unlabeled). This why we have the need to label some data based on our linguistic knowledge to be able to check how good nltk-model is.

a. accuracy of TreeBank tagger

Before removing functional words, we labeled 10 sentences. We let the Treebank-tagger, given by nltk, predict the tags of these sentence, and we compared between the predicted tags with the labeled data. Based on that we found out the accuracy of the tagger which is 82%.

There were no need to check the accuracy of wornet_tagger because it is straight-forward, and there is no room for mistake.

b. accuracy of WordNet lemmatizer

During lemmatizing process, our model drops out stop words, like propositions, pronouns and so on. The reason is they give not information, and cannot be lemmatized because they will never have affixes/extensions.

-Based on our linguistic knowledge, we labeled first 30 sentences to use them for measuring the accuracy of the lemmatizer. Her are the results:

-accuracy of our lemmatizer (without tagging) is approximately: 84%

-accuracy of our lemmatizer (with tagging) is approximately: 97%'

These results show how important tagging is,in additional to cleaning data, for optimizing the lemmatizer.

iv. Threshold

Because languages have an amazing complex phenomenon, there is no agreement among linguistic institutions about how many words an average (American) native English speaker uses on daily basis. But the range is between 7000-35000 words or lemmas. According to the Oxford English Corpus, native speakers use just 7000 words for 90% of everything they say and write! [6]

We decided to consider the least number (=7000 lemmas) as threshold for our data (*figure 1*). The reason is that we want to avoid overfitting problem. We want to draw your attention again to that **one** of our main goals is to let English-learners learn least possible vocabularies which helps him or her to speak English as natural as possible, not make him or her on the same level as a native speaker regarding number of vocabularies. Regarding the **phrasal verbs**, we found no study or statistics which can help to decide how many phrasal verbs an English-language learner needs to speak English fluently (around 90% of what a native speak does). We thought about to approaches. The first one is mapping among the verbs which appear in the 7000 most frequent lemmas and the verbs which appear in the phrasal verbs. The second approach is to consider the border that separates the phrasal verbs which occur just once from the ones which occur more than one in the series. For simplicity we will go with the second approach (*figure 2*).

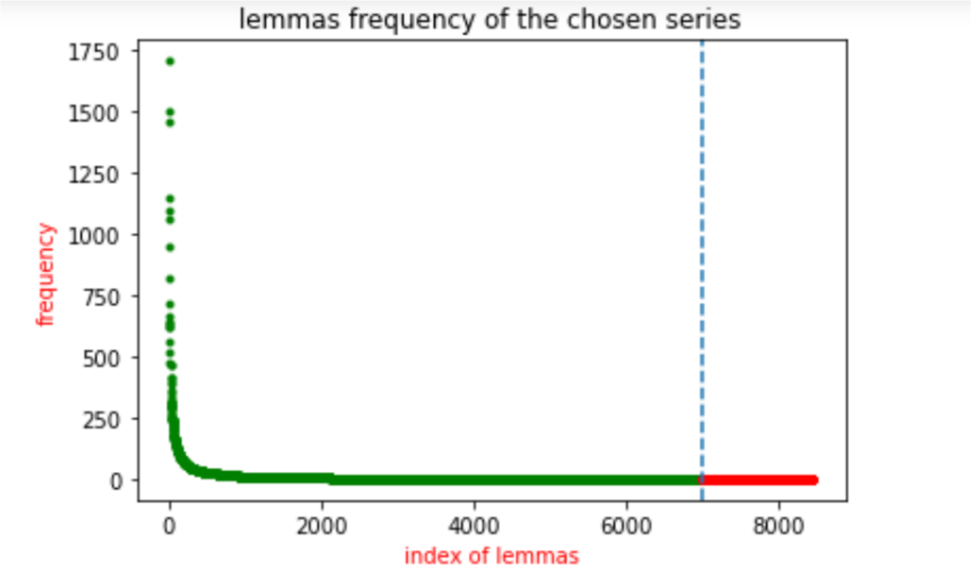


Figure1

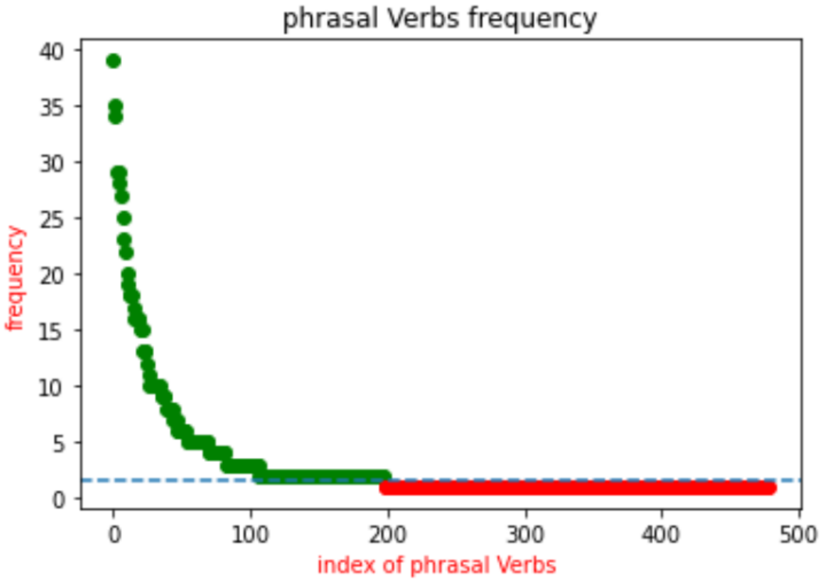


Figure2

The L-shape of both *figure1*, and *figure2* is somehow expected because our model has sorted the lemmas and phrasal verbs before the visualizing. It is clear that the frequency values in *figure1* increase from the left side to right dramatically. This means it has extreme values (z-score can be used here to measure how extreme they are). But the frequency values in *figure1* get suddenly stabilize after it reach about 25. While in figure 2 the values tend to be less extreme and stabilize gradually after they reach the frequency value

2. The density of lemmas’ frequency (*figure1*) is a little unclear because of the wide spread of these values, and because of the extreme values. To be able to see the dispersion of data we need to use boxplot.

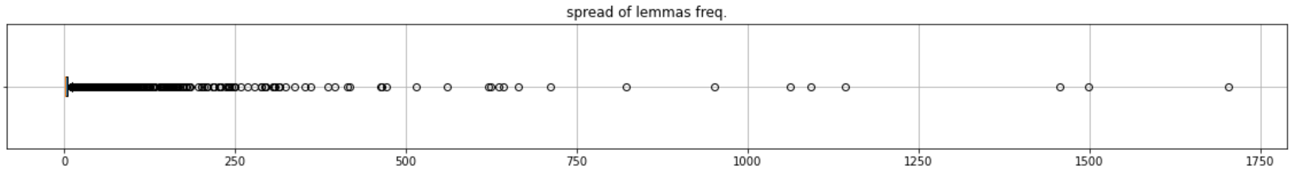


Figure3

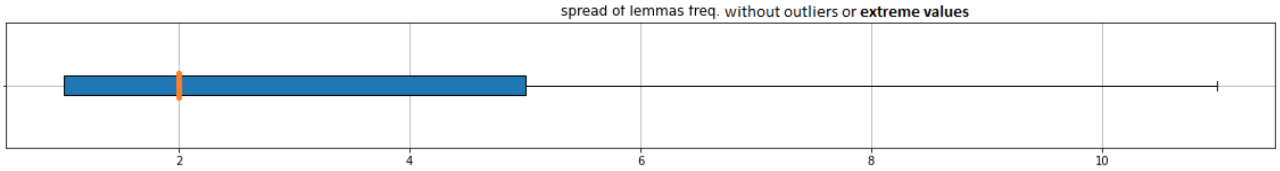


Figure4

Generally speaking, the main density of lemmas is in range of 1-250 as *figure3* shows and after 500 the density begins to reduce significantly. But if we zoon in the data by removing extreme values as we can see in *figure4*, we will get different insights. Like for example 25% of lemmas occurs once in the series while 50% of lemmas occur between 1 and 5 times in the data set. The mean of the frequency values is 2, whiskers positions in range 5 to 15, and the Inter-quartile of lemmas frequency is 4. The main advantage of it that it is not affected by extreme values.

D. Recommendations

At the beginning, we thought to use sequence models, naïve base or/and N-grams to extract the phrasal verbs from series but we found a way which is simpler and computationally efficient. Anyway, we might need to use one of them to extract phrasal verbs which consists of more than two words. In other words, our model extracts the first two words of a phrasal verb, no matter whether the phrasal verb consists of two or more words. If we had enough time, we would also work on other kinds of phrases which their meanings are different from what the individual words might suggest, like idioms. Another thing we would use more time on is data cleaning because it is endless process, especially when dealing with data from natural languages. Labeling more data is not less important than other things mention early. This is because it helps us in getting a more precise validation of the model. It is also a good idea to add more series to the existing ones to reach better results. In addition, we might need to consider series from other English-speaking countries like Britain, Canada, and Australia, not just American ones. At the end, when model optimized as possible, we have a plan to implement this model on Norwegian and possibly on some of the most widely spoken languages in the world, taking in consideration the syntactical and morphological structure of every language.

DI. Resources:

1. Series

   www.OpenSubtitles.org

   www.Subscene.com

   - A list of the used series' names:
     - superman of tomorrow
     - I love Lucy
     - life to tell
     - high school musical
     - sex and the city
     - imperfect
     - avocado toast
     - world on fire

2. nltk-model

   https://www.nltk.org/

3. Treebank pos tags list

   https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.htmL

4. wordnet pos tags list

   https://wordnet.princeton.edu/documentation/wndb5wn

5. Link to the data set
   https://drive.google.com/file/d/16oimMLL8BP9TxfO9_LsUpHpyMB9Erbq9/view?usp=sharing

6. -A study from The Oxford English Corpus about number of English words we need to learn

   http://www.englishteachermelanie.com/study-tip-the-english-words-you-need-to-know/

   -Number words a native speakers use:

   http://testyourvocab.com/blog/2013-05-08-Native-speakers-in-greater-detail#newMainchartNative

7. A primitive method borrowed from sackoverflow-website

   https://stackoverflow.com/questions/15586721/wordnet-lemmatization-and-pos-tagging-in-python

8. speech corpora
   https://en.wikipedia.org/wiki/Speech_corpus