MSC NATURAL LANGUAGE PROCESSING 2022-2023

UE 803 - DATA SCIENCE

Analysis of Wikipedia articles using Spacy and Stanza

Students:
Prunelle DAUDRÉ–TREUIL
Ekaterina GOLIAKOVA
Inés HERNÁNDEZ

Introduction

In this paper, we will investigate two different classes of Wikipedia articles: Writers and Astronauts. In the first section, we will discuss the data collection process: how data was obtained, pre-cleaned, and stored.

In the second section, we will compare the article classes among themselves: how a certain article category affects the length of an article, and what named entities are more frequent in what article class. In addition to that, we will also analyze the vocabulary and compare different methods for obtaining the top frequent words (*Wordcloud* and counting *Spacy* token frequencies). Finally, we will have a look at different classification models for these two categories of texts and analyze how different approaches to text-pre-processing affect classification accuracy.

In the third section, we will move from comparing article classes to comparing different tokenization libraries: *Spacy*[1] and *Stanza*[2]. We will have a look at the way these libraries perform sentence segmentation and tokenization and compare the Part of Speech (PoS) tagging performed by each of them.

1 Data Collection

Given the task to construct a corpus containing two different classes of Wikipedia articles, we have analyzed how the Wikipedia articles are structured and noticed that both for Writers and Astronauts, there exists a "Category" page[34]. However, we also noticed that each of these category pages contains multiple different subcategories, that group different writers or astronauts. Since a writer or an astronaut can be potentially a part of many such sublists, we made a decision to work with the subcategories by nationality[56] only which ideally should have little overlap and should allow us more efficient scrapping where the same article is not scrapped multiple times. However, our chosen subcategories have each a list of subcategories of their own (a list for each nationality), therefore, our scrapping plan was the following:

- Take the subcategories Writers by nationality and Astronauts by nationality as the base
- Take all of its subcategories (lists of articles for each nationality)
- Filter out non-nationality-related lists
- For each nationality list extract the list of articles for personalities

[1] https://spacy.io/

[2] https://stanfordnlp.github.io/stanza/

[3] https://en.wikipedia.org/wiki/Category:Writers

[4] https://en.wikipedia.org/wiki/Category:Astronauts

[5] https://en.wikipedia.org/wiki/Category:Writers_by_nationality

[6] https://en.wikipedia.org/wiki/Category:Astronauts_by_nationality

- Extract text for each personality

Since the task involved working and filtering subcategories of Wikipedia, we have chosen to work with *WikipediaAPI* module which allows easy access to Wikipedia categories and accessing the articles by category[7].

Following the plan above, we obtained a list of articles that were grouped into two folders: Writers and Astronauts. Before moving to the next tasks we did pre-cleaning: many of the texts contained multiple linebreaks, non-punctuation special symbols like (< , >, = etc.), tabulation, or multiple spaces. During the pre-cleaning process, all of this was removed and replaced with a single space.

After the cleaning and collection were completed, we obtained: 11548 articles for Writers and 451 articles for Astronauts. Due to the unbalanced number of articles, we selected a random sample of 200 articles for writers and 200 articles for astronauts, totaling 400 articles. This sample was used for tasks of Sections 2.1-2.3. For the classification task of Section 2.4 we created another, slightly bigger, random sample of total of 800 articles (400 for Writers and 400 for Astronauts).

In addition to the collected texts from Wikipedia, for the tasks of Section 3, we also added a function that allows us to extract text, segment/tokenize it for any web page, using BeautifulSoup. The same pre-cleaning process as described above, is applied when using this function. For the tasks of Section 3, we have selected another sample of 100 articles, however, just across the Writers category.

2 Data Analysis

In this section, we will compare the tokenization and classification results between two categories of articles we stored: Writers and Astronauts.

2.1 Sentence and token counts

As the first step, we decided to take a closer look at the content of the articles selected, and for this, we split each of them into sentences using *Spacy*. On average, the articles belonging to the category Astronauts have a higher number of sentences (43.98 versus 30.69 for Writers). As for the maximum amount of sentences per article, we observed 324 sentences for Astronauts and 267 sentences for Writers. As for the minimum, we obtained 2 sentences per article for both categories. As shown in figure 1 the majority of articles have between 2 and 30 sentences approximately, while very few articles have more than 100 sentences.

For tokenization, we also used *Spacy* and in this process, we removed stopwords and punctuation. First, we tokenized each sentence of each article, and then we joined all these tokens to get an overall count of tokens per category. Regarding the number of tokens per sentence, both categories showed similar results in the average amount: 12.8 tokens for Astronauts and 11.5 tokens for Writers. The maximum number of tokens per sentence is

[7] https://wikipedia-api.readthedocs.io/en/latest/API.html

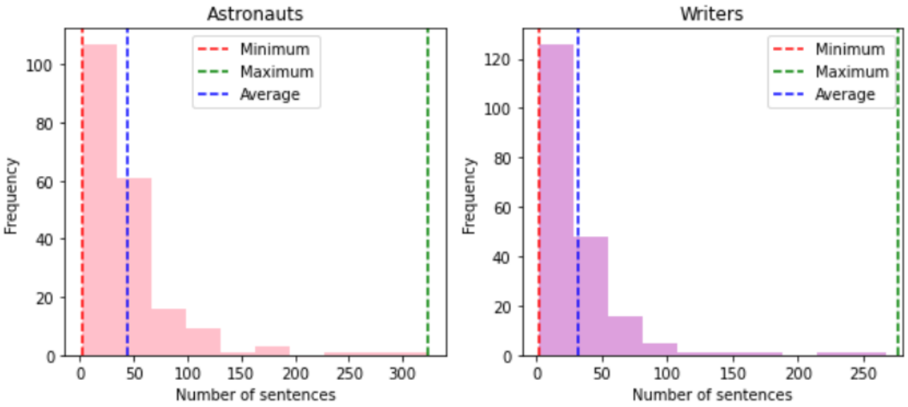


Figure 1: Sentences per article for both categories.

slightly higher in the Writers articles (177 vs 152 for the Astronauts). The minimum for both is 0 that is explained by our sentence tokenization process which included removing stopwords. As shown in figure 2, most sentences from the Writers category have between 0 and 25 tokens, and Astronauts' sentences have between 0 and 30 tokens.

The next step was counting tokens per category, joining all the tokens of all sentences. On average, Astronauts articles have more tokens (557.62) than Writers articles (354.4)[8], with the maximum amount of tokens for Writers is 3818, and the minimum is 13; while for Astronauts the maximum is 4449 and the minimum is 29.

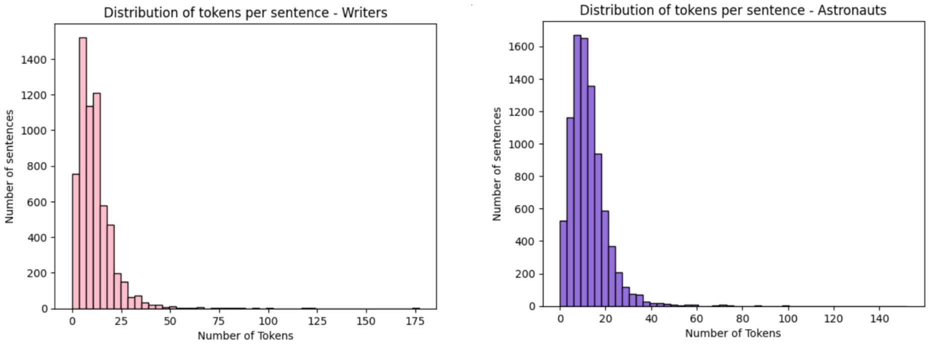


Figure 2: The number of tokens per sentence for all articles.

[8] We have also confirmed this using a t-test, and obtained the following values: t-statistic = 4.097; p-value = 0.0000507.

2.2 Named Entities Recognition

We continued the investigation, extracting named entities using *Spacy* library in order to compare the article classes: whether or not some of the named entities are more frequent in a particular article class. *Spacy* has 18 entity types as shown in figure 3. The entity PERCENT is the least frequent in both types of articles and ORG the most frequent. We can also quickly observe that there is a bigger number of entities in the Astronauts articles ORG, DATE and PERSON are the most common entities, with a big difference to Writers.

There are almost 8000 ORG entities recognized in Astronauts articles, while in Writers articles the number is a little over 4000, almost half. The entity recognized in both categories nearly the same amount of times was NORP (Nationalities or religious or political groups). It is also relevant that the entities WORK OF ART and LANGUAGE are the only ones that appear more times in the Writers articles, given the nature of the content of this category[9].

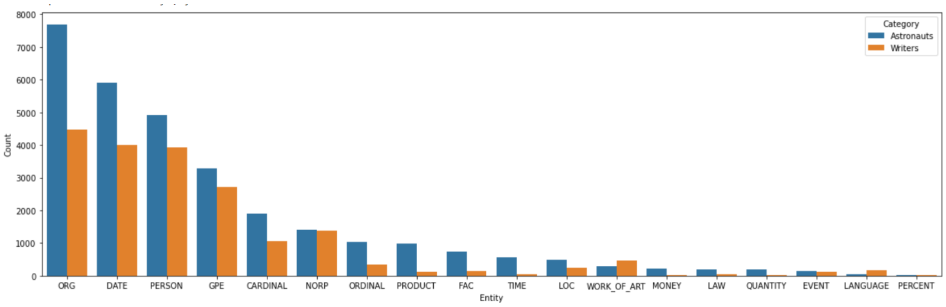


Figure 3: Comparision of Named Entities recognized in each category.

2.3 Vocabulary

This section's focus is on the type of words used in both categories. To do so, we compared *WordCloud*[10] library's top of frequent words with our own created by ranking *Spacy* tokens by the number of their occurrences.

First, we joined all of our articles in two strings, one for each category, to compute the frequency of words in the whole corpus. We then tokenized them using *Spacy*, excluded the punctuation, stop words, and all tokens containing only blank characters. With that, we constructed a DataFrame of all 50 most frequent tokens by category and their number of occurrences. You will be able to see the results of this work in Appendix's Figures 8 and 9.

After generating the two wordclouds (see Figure 4 and 5), we were able to compare our results with it. By doing this, we found that 80% of our list of frequent tokens for the Writers category was present in *WordCloud* frequent list as well, however, only 52% was shared for the Astronauts category.

[9] This was confirmed using a t-test, obtained values for WORK_OF_ART: t-statistic = 2.8699, p-value = 0.00433; for LANGUAGE: t-statistic = 2.8539, p-value = 0.00454.

[10] http://amueller.github.io/word_cloud/

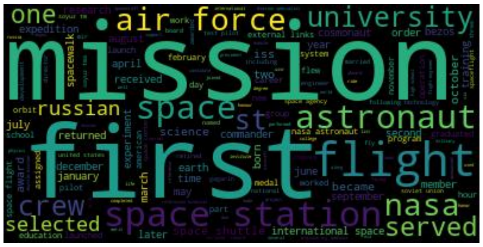


Figure 4: Wordcloud for the Astronauts corpus.

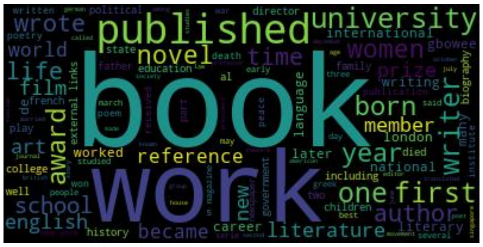


Figure 5: Wordcloud for the Writers corpus.

We continued the investigation of what type of words appear uniquely in one list or the other. You can see examples of words that appeared in just our top or just Wordcloud top in Figures 6 and 7.

We noticed that for the Writers list, our top contained some plural forms that do not appear in the Wordcloud, probably because the library recognizes both the singular and the plural as the same lemma. Writers wordcloud also contains words like 'one', 'many' and 's' that were likely filtered by our use of the *Spacy* tokenization. Interestingly, our top doesn't include 'art' which is an important term for 'Writers' articles, it may be connected to the fact that having treated plurals and singulars as different forms we missed out on 'art' which appeared below the top 50.

`peace 2011 war early books years series works isbn references`

(a) Unique for our list

`part art s many became reference first including play one`

(b) Unique for Wordcloud

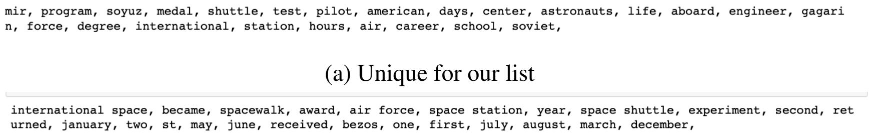
Figure 6: Words that appeared uniquely in Wordcloud library top 50 or uniquely in top 50, generated by selecting the most frequent Spacy tokens. Writers category.

For the Astronauts list, we can see that *Wordcloud* tokenizes 'international space' as one token and we can see in our list 'international' appears as a separate token. The same can also explain words like 'station' appearing only in our top but not in Wordcloud top. It seems like the *Spacy* tokenization doesn't tokenize complex proper nouns like the International Space Station as one token, therefore, during ranking we lose the connection.

Overall, we can notice that Wordcloud works with lemmas and co-occurrences while with Spacy we're treating plurals and singular forms as different cases, which may lead to missing some important information, like Art" for the Writers category.

2.4 Classification

The next task was to train a classifier in order to label an article either as Writers or Astronauts. We also decided it would be interesting to see the effect of tokenization of the text on the classification result an for this performed the classification on the following types of



```
mir, program, soyuz, medal, shuttle, test, pilot, american, days, center, astronauts, life, aboard, engineer, gagari
n, force, degree, international, station, hours, air, career, school, soviet,
```

(a) Unique for our list

```
international space, became, spacewalk, award, air force, space station, year, space shuttle, experiment, second, ret
urned, january, two, st, may, june, received, bezos, one, first, july, august, march, december,
```

(b) Unique for Wordcloud

Figure 7: Words that appeared uniquely in Wordcloud library top 50 or uniquely in top 50, generated by selecting the most frequent Spacy tokens. Astronauts category.

text tokenization:

- Punctuation and stopwords removed (the same logic we used for the tokenization and segmentation)
- Removing the most frequent 50 words for each category
- Not filtering any tokens

To compare the effect of each of the tokenization types we ran the classifiers only for one iteration.

Prior to the classification all texts were lowerecased and processed in one of the ways described above. As for the classification models, we chose Perceptron and SVM.

For the training dataset, we loaded 800 articles[11], shuffled it, encoded the labels to turn them into 0 and 1, performed the needed tokenization and split the corpus into train and test (80% and 20% respectively). After, we used *sklearn*'s TfidfVectorizer to vectorize the data and trained the model with 1 iteration.

Tokenization method	**Perceptron**	**SVM**
Removing stopwords and punctuation	0.98	0.85
Removing top 50 frequent words	0.93	0.52
No filtering	0.985	0.66

Table 1: Accuracy of classification of Writers and Astronauts texts after 1 iteration.

As can be seen in Table 1, the accuracy of Perceptron remained very high even when removing important tokens or leaving the noise in. As for SVM, we can see the effect of different tokenization more clearly: removing frequent tokens drops the accuracy down for around 40% and while the drop for non-filtered texts is less drastic, it's nevertheless significant.

To explain the high performance of Perceptron even in bad conditions we looked at our data and found out that only 28% of Writers vocabulary is shared with Astronauts, which can explain high accuracy even with no pre-processing or removing popular words.

[11] These were randomly selected from our previously scrapped data: 400 for Writers and 400 for Astronauts

3 Spacy and Stanza comparison

In this section, we will cover the experiments regarding the comparison of different tokenization libraries and for this, we have chosen *Spacy* and *Stanza*. As mentioned before, for the experiments we have chosen a sample of 100 articles from the Writers category. All the segmentation, tokenization and PoS tagging experiments were run on the same set of text and tokens. One of the most evident differences between the libraries is the runtime: *Stanza* tokenization can take upwards of 10 times longer than *Spacy* tokenization. In this paper, however, we won't provide the runtime comparison but for easier re-running of the code, all *Stanza* tokenization was stored in files once completed.

3.1 Segmentation

Our first experiment is to compare sentence segmentation per article, using both libraries. In Table 2, we can notice that it appears that *Spacy* tokenizes the text on average into more sentences than *Stanza* does. We also confirmed this theory using a t-test and the values we obtained[12] allowed us to reject the null hypothesis and accept the hypothesis that on average *Spacy* tokenizes text into more sentences than *Stanza* does.

Metric	**Spacy**	**Stanza**
Average	30.8	28.2
Median	25	21.5
Std.	26.2	24.4
Min	3	3
Max	155	133
Total	3077	2823

Table 2: Metrics describing the distribution of the number of sentences, segmented per article for Spacy and Stanza.

However, when we looked into different cases of sentence segmentation we noticed several groups of cases where *Spacy* incorrectly produces fewer sentences than *Stanza* does, as well as when *Spacy* correctly produces fewer sentences than *Stanza*. We can have a look at the examples of different sentence tokenization for *Spacy* and *Stanza* in Table 3, overall we can see that *Stanza* frequently separates sentences when there is a capital letter in the middle of a sentence which leads to correct segmentation when a title is not followed by any punctuation (issue 2) and incorrect segmentation when there is a named entity in a sentence (issue 3). We can also see that *Stanza* deals better with incorrect punctuation (e.g., missing space before a period), however, in all the issues (1)-(3) *Stanza* is the one producing more sentences, while we know that overall *Stanza* produces fewer sentences. Therefore, we can theorize that issue (4) is actually the most occurent for our set of articles about writers.

Having filtered out all the non-matching sentences, we were left with a total of 2263 (73% of *Spacy* and 80% of *Stanza* segmentation) sentences that were tokenized in the same way across all our original 100 texts.

[12] t-statistic = 4.3685; p-value = 0.000031

Issue type	Spacy sentence	Stanza sentence
(1) No space before punctuation	He successfully underwent a ground-breaking bilateral arm transplant in August 2016.Peck	He successfully underwent a ground-breaking bilateral arm transplant in August 2016.
(2) No punctuation separating title	Life He was a son of Julius Sabbe and the eldest of seven children.	Life
(3) Incorrect segmentation before capital letter	Separated from Dave in a bustling street, she fell and twisted her ankle ...	Separated from
(4) Segmentation with a lot of non-English tokens	Works in English From Wŏnso Pond (Feminist Press 2009)	Works in English From Wŏnso Pond (Feminist Press 2009) ISBN 978-1-55861-601-1

Table 3: Examples of different sentence segmentation generated by Spacy and Stanza.

	Token count
Spacy vocabulary size	8488
Stanza vocabulary size	8526
Shared vocabulary	8264

Table 4: Vocabulary size across the shared sentences.

3.2 Tokenization

Using the 2263 sentences obtained in Section 3.1, we continued the experiments and compared the tokenization. Each of the shared sentences was treated as a string, lowered prior to further tokenization, and then processed by *Spacy* and *Stanza* to be separated into tokens.

Our first experiment regarding tokenization was the vocabulary size: how many unique tokens were produced by both libraries which is shown in Table 4. As we can see 96%-97% of tokens were tokenized in the same way Looking closely into tokens that were split differently, we noticed that the majority of them are non-English tokens or numbers, however, we also noticed several interesting differences for English tokens:

- *Spacy* and Stanza have different approaches to hyphen-separated words, such as Anti-colonialism", "multi-awardee", "non-denominational" and others. *Stanza* treats such words as one token and *Spacy* separates them into two or more.
- Similarly, *Stanza* tokenizes possessive "'s (or an auxiliary verb) together with a noun, while Spacy treats such cases as several tokens. For example, "unesco's" is one token for *Stanza* and two separate ones for *Spacy*.

The next experiment was dedicated to finding all occurrences of a wordform (non-unique tokens) that were tokenized in the same way by both *Spacy* and *Stanza* in all of the shared sentences found in Section 3.1. The algorithm for finding such occurrences is the following:

- Find all occurrences of a wordform tokenized in the same way by both libraries

- Since as we saw above, the number of occurrences of a wordform can be different (e.g. *Stanza*'s tokenization would stitch "'s" to a noun, and *Spacy* would have the noun separately), find the smallest number of occurrences between the two libraries (*k*)

- Return first *k* occurrences of a wordform for both *Spacy* and *Stanza*.

Note that this algorithm doesn't take into consideration that some of the occurrences can actually be happening in different sentences. When treating all of the shared sentences as a single string, non-separated into sentences we matched 48070 tokens (approx. 99% of all tokens generated for both *Spacy* and *Stanza*). However, when we slightly modified the algorithm and matched only wordforms occurring within the same sentence and then combine the data for all sentences, we found only 41443 shared tokens (approx. 85% of all tokens for both *Spacy* and *Stanza*).

All of the 41443 tokens were stored with the PoS information from their original sentence and prepared for the analysis of PoS annotation by the libraries.

3.3 PoS tagging

For this section, we began by finding all unique UPOS tags and both *Spacy* and *Stanza*. After checking that the two libraries use the same set of tags, we created a DataFrame using *Spacy* tags as columns and *Stanza* tags as columns. Using this DataFrame, we calculated frequencies of all UPOS pairs (e.g. NOUN x ADJ, number of times a *Spacy* NOUN token was recognized as ADJ). For most of UPOS, both libraries agree in more than 85% of cases, however, there seemed to be a lot of disagreement for X, INTJ and PROPN (full table can be found in Appendix Figure 10).

First, we decided to investigate closer popular UPOS and what is the main disagreement on them, for this, we created a pie chart showing the PoS tags used by *Stanza* for the following *Spacy*'s tags: NOUN, VERB, ADJ, ADV.

When looking at how *Spacy*'s noun tags were recognized by *Stanza*, we can see that 88% of tokens tagged as a common noun by *Spacy* are also tagged common noun by *Stanza* (can be seen in Appendix Figure 11). When the two libraries disagree, the token is mainly tagged as PROPN (7.69%) by *Stanza*, the two libraries seemingly in disagreement over the distinction between common nouns and proper nouns. The second tag used by *Stanza* is X (2.45%), a tag used when the UPOS is not recognized, like for foreign words or gibberish, which we will get back later to.

When investigating the tags for VERB tokens of *Spacy*, we noticed that in 93% of the tokens the two libraries found the same UPOS (can be found in Appendix 12). The main disagreement between them is for nouns (2.59%), proper nouns (1.81%), and adjectives (1.08%). We can note that this time, category X is pretty low at 0.94%.

Continuing the experiment, we have looked into *Spacy*'s adjective tags, and 86% of them are similar between the two libraries (see Appendix 13). In case of disagreement, the two main tags given by *Stanza* are Noun (4.86%) and Proper Noun (4.37%). Overall, the confusion between adjectives and nouns for the libraries seems to be much higher than for verbs: *Spacy*'s verb are confused with 4 other PoS by *Stanza*, while nouns and adjectives are confused with 7 and 10 other UPOS respectively.

As mentioned above, there seemed to be a big disagreement between libraries on PROPN and X tags and we decided to investigate this further. We found out that for there 14 tokens that were tagged X that were recognized as a PoS by *Stanza*, while there were 952 tokens that were tagged X by *Stanza* but recognized as a PoS by *Spacy*. It seems like *Spacy* recognizes the PoS of tokens more often. We decided to investigate what kind of words *Spacy* and *Stanza* each put in the X category. For *Spacy*, we find some named entities (emmanuel, Newcastle ...), numbers, and a few foreign words. Since *Stanza* tagged a lot of tokens as X (over 900), we choose to examine a random selection of words. In this random selection, we could see mainly foreign words but also some more strange occurrences like the token 'golf'. A look at the context of the word, we found that it was used in a sentence in Dutch, therefore seeing it as a foreign word can be seen as more logical. This same occurrence of "golf" was tagged as proper noun by *Spacy*. Having looked at another token in the context of a sentence in a foreign language, 'geographischen', we noticed that if *Stanza* put it in the X category, *Spacy* put it in the PROPN. So, we can conclude that if *Spacy* gives a UPOS tag to tokens more often, it doesn't mean that it is this is neccessary correct for the context of foreign words.

4 Conclusion

In our work, we compared different classes of texts (Writers and Astronauts) and different tokenization libraries. We have found that Astronauts on average have longer articles and more tokens per article and despite that certain named entities like WORK_OF_ART and LANGUAGE are more frequent in Writer" texts. We also found that both text classes have significantly different vocabulary which the classification accuracy of the Perceptron model very high despite using very noisy data. After that, we focused on Writers texts only and compared different tokenization libraries (*Spacy* and *Stanza*). We have found that *Stanza* has certain issues with sentence segmentation before a capital letter, however, still produced fewer sentences than *Spacy*. *Spacy*, on the other side, splits complex words like "neo-colonial" into several tokens, which creates fewer unique tokens and leads to a smaller vocabulary size. As for PoS, we have found that there is a big disagreement between libraries on tagging foreign words in a context of a foreign sentence. Overall, *Spacy* seems to have a tendency to attempt to tag any (even previously unseen) word with a PoS tag which can lead to incorrect tagging, while for questionable cases *Stanza* is more prone to mark a PoS as unrecognized (tag X).

Contents

1 Data Collection 1

2 Data Analysis 2

2.1 Sentence and token counts 2

2.2 Named Entities Recognition 4

2.3 Vocabulary 4

2.4 Classification 5

3 Spacy and Stanza comparison 7

3.1 Segmentation 7

3.2 Tokenization 8

3.3 PoS tagging 9

4 Conclusion 10

List of Figures

1 Sentences per article for both categories. 3

2 The number of tokens per sentence for all articles. 3

3 Comparision of Named Entities recognized in each category. 4

4 Wordcloud for the Astronauts corpus. 5

5 Wordcloud for the Writers corpus. 5

6 Words that appeared uniquely in Wordcloud library top 50 or uniquely in top 50, generated by selecting the most frequent Spacy tokens. Writers category. 5

7 Words that appeared uniquely in Wordcloud library top 50 or uniquely in top 50, generated by selecting the most frequent Spacy tokens. Astronauts category. 6

8 Top 50 most frequent Spacy tokens for Astronauts category. 13

9 Top 50 most frequent Spacy tokens for Writers category. 14

10 Frequency of Stanza UPOS corresponding to a Spacy UPOS. The columns represent Spacy tags and rows - Stanza's. On each intersection, you will find the percent of Spacy tokens from a given column interpreted as Stanza tokens of a given row. 15

11 Stanza's UPOS corresponding to Spacy's NOUN. 16

12 Stanza's UPOS corresponding to Spacy's VERB. 17

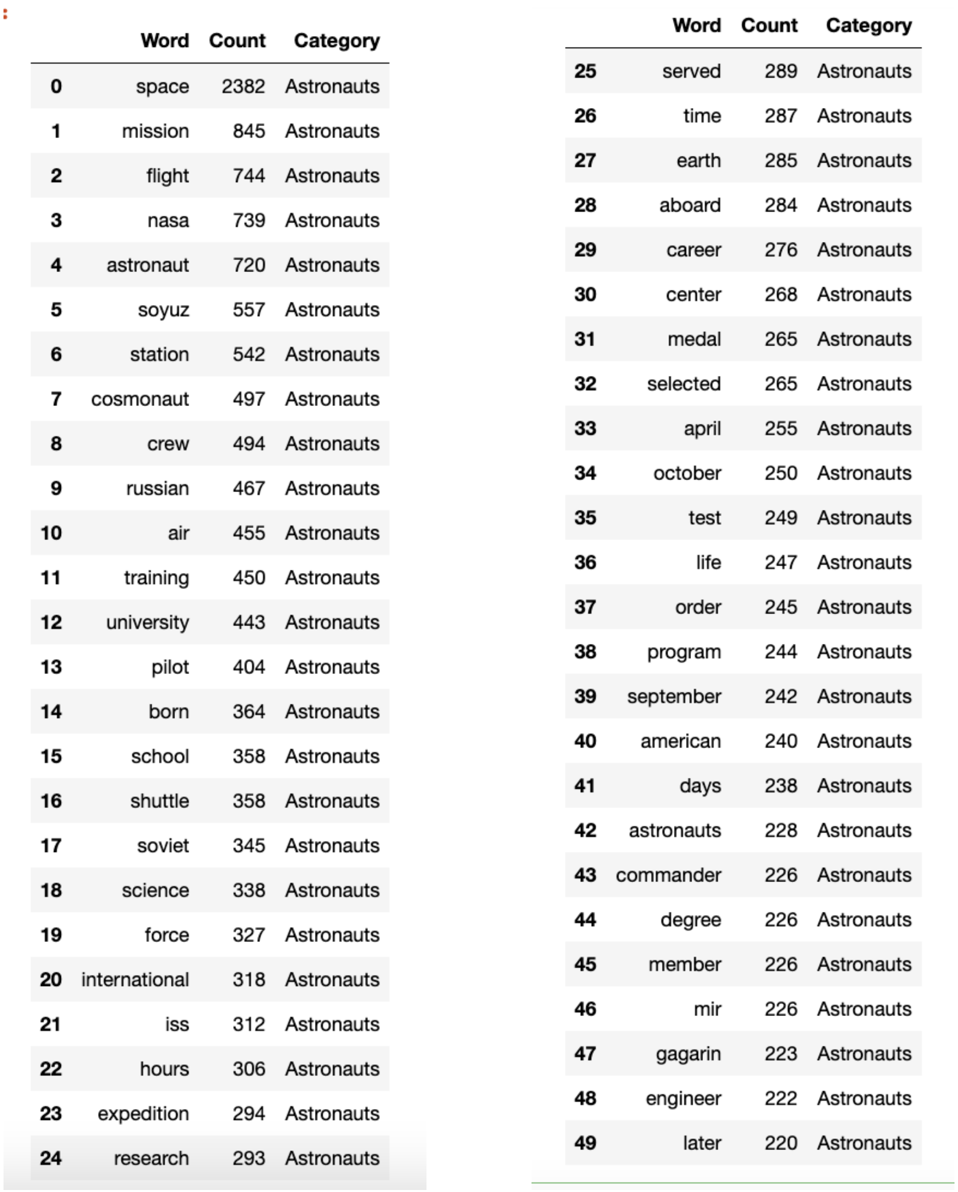
13 Stanza's UPOS corresponding to Spacy's ADJ. 18

14 Stanza's UPOS corresponding to Spacy's ADV. 19

List of Tables

1 Accuracy of classification of Writers and Astronauts texts after 1 iteration. . 6
2 Metrics describing the distribution of the number of sentences, segmented per article for Spacy and Stanza. 7
3 Examples of different sentence segmentation generated by Spacy and Stanza. 8
4 Vocabulary size across the shared sentences. 8

Annexes



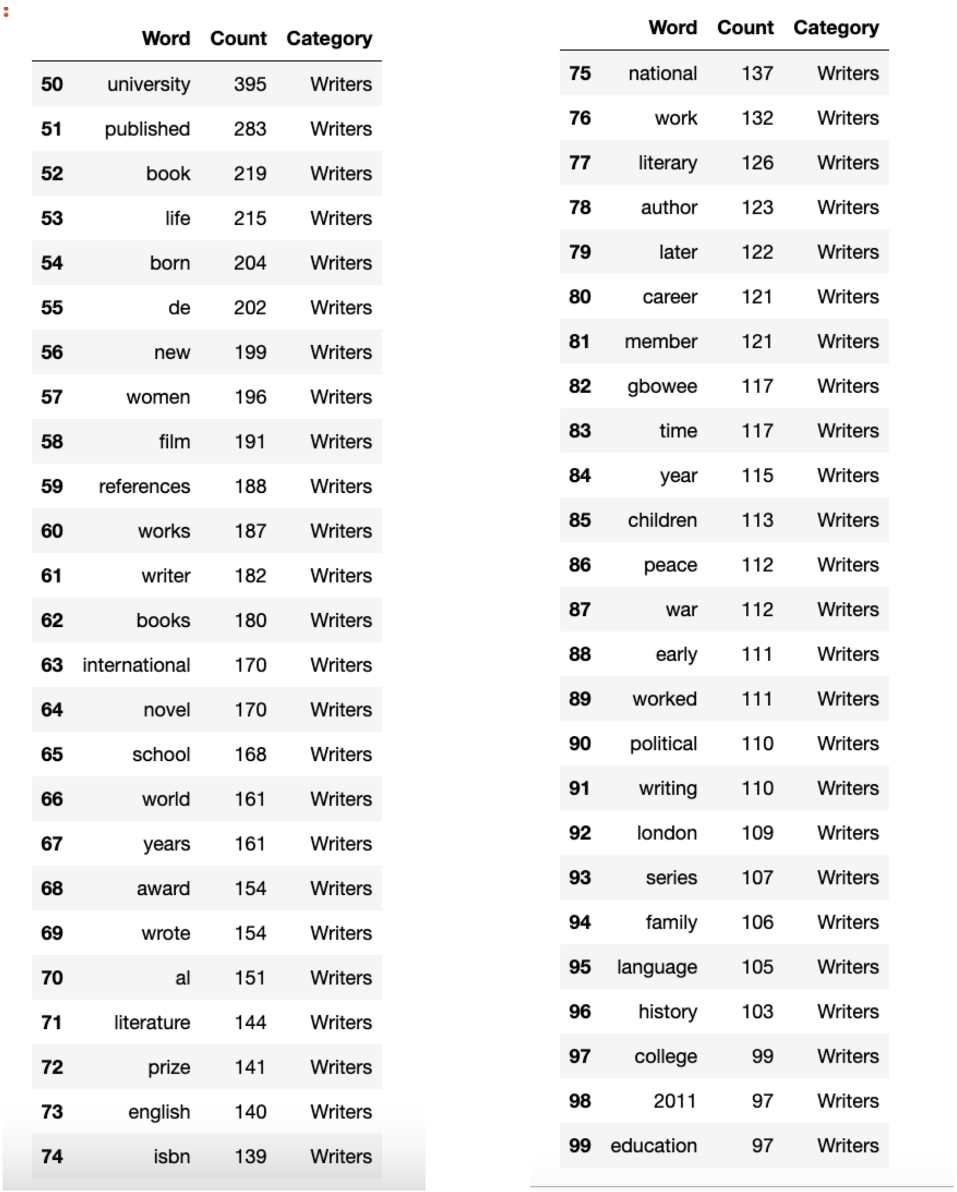
	Word	Count	Category
0	space	2382	Astronauts
1	mission	845	Astronauts
2	flight	744	Astronauts
3	nasa	739	Astronauts
4	astronaut	720	Astronauts
5	soyuz	557	Astronauts
6	station	542	Astronauts
7	cosmonaut	497	Astronauts
8	crew	494	Astronauts
9	russian	467	Astronauts
10	air	455	Astronauts
11	training	450	Astronauts
12	university	443	Astronauts
13	pilot	404	Astronauts
14	born	364	Astronauts
15	school	358	Astronauts
16	shuttle	358	Astronauts
17	soviet	345	Astronauts
18	science	338	Astronauts
19	force	327	Astronauts
20	international	318	Astronauts
21	iss	312	Astronauts
22	hours	306	Astronauts
23	expedition	294	Astronauts
24	research	293	Astronauts

(a) Top 1-25

	Word	Count	Category
25	served	289	Astronauts
26	time	287	Astronauts
27	earth	285	Astronauts
28	aboard	284	Astronauts
29	career	276	Astronauts
30	center	268	Astronauts
31	medal	265	Astronauts
32	selected	265	Astronauts
33	april	255	Astronauts
34	october	250	Astronauts
35	test	249	Astronauts
36	life	247	Astronauts
37	order	245	Astronauts
38	program	244	Astronauts
39	september	242	Astronauts
40	american	240	Astronauts
41	days	238	Astronauts
42	astronauts	228	Astronauts
43	commander	226	Astronauts
44	degree	226	Astronauts
45	member	226	Astronauts
46	mir	226	Astronauts
47	gagarin	223	Astronauts
48	engineer	222	Astronauts
49	later	220	Astronauts

(b) Top 26-50

Figure 8: Top 50 most frequent Spacy tokens for Astronauts category.



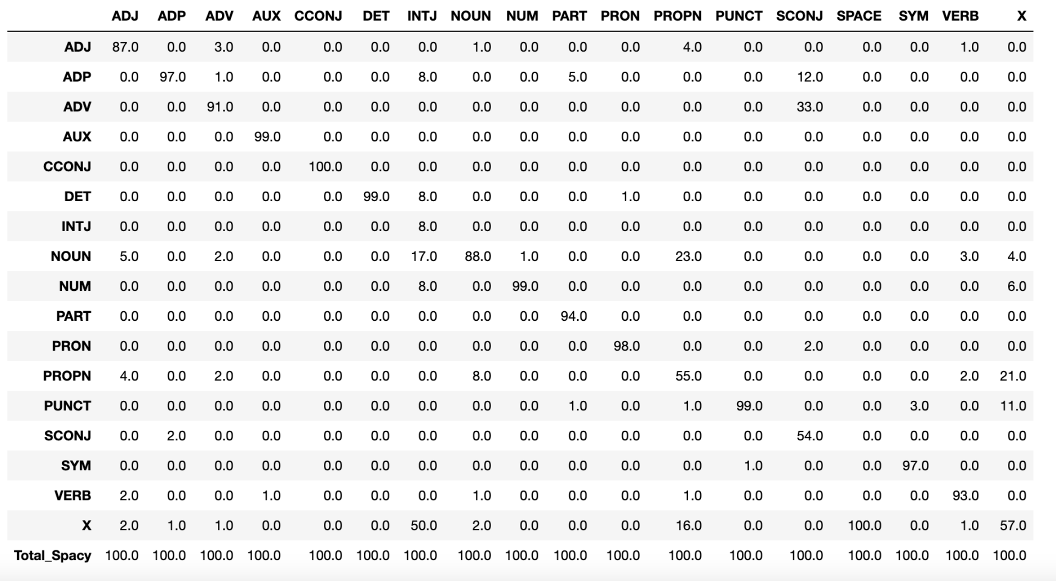
	Word	Count	Category
50	university	395	Writers
51	published	283	Writers
52	book	219	Writers
53	life	215	Writers
54	born	204	Writers
55	de	202	Writers
56	new	199	Writers
57	women	196	Writers
58	film	191	Writers
59	references	188	Writers
60	works	187	Writers
61	writer	182	Writers
62	books	180	Writers
63	international	170	Writers
64	novel	170	Writers
65	school	168	Writers
66	world	161	Writers
67	years	161	Writers
68	award	154	Writers
69	wrote	154	Writers
70	al	151	Writers
71	literature	144	Writers
72	prize	141	Writers
73	english	140	Writers
74	isbn	139	Writers

(a) Top 1-25

	Word	Count	Category
75	national	137	Writers
76	work	132	Writers
77	literary	126	Writers
78	author	123	Writers
79	later	122	Writers
80	career	121	Writers
81	member	121	Writers
82	gbowee	117	Writers
83	time	117	Writers
84	year	115	Writers
85	children	113	Writers
86	peace	112	Writers
87	war	112	Writers
88	early	111	Writers
89	worked	111	Writers
90	political	110	Writers
91	writing	110	Writers
92	london	109	Writers
93	series	107	Writers
94	family	106	Writers
95	language	105	Writers
96	history	103	Writers
97	college	99	Writers
98	2011	97	Writers
99	education	97	Writers

(b) Top 26-50

Figure 9: Top 50 most frequent Spacy tokens for Writers category.



	ADJ	ADP	ADV	AUX	CCONJ	DET	INTJ	NOUN	NUM	PART	PRON	PROPN	PUNCT	SCONJ	SPACE	SYM	VERB	X
ADJ	87.0	0.0	3.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	1.0	0.0
ADP	0.0	97.0	1.0	0.0	0.0	0.0	8.0	0.0	0.0	5.0	0.0	0.0	0.0	12.0	0.0	0.0	0.0	0.0
ADV	0.0	0.0	91.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	33.0	0.0	0.0	0.0	0.0
AUX	0.0	0.0	0.0	99.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CCONJ	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DET	0.0	0.0	0.0	0.0	0.0	99.0	8.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
INTJ	0.0	0.0	0.0	0.0	0.0	0.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NOUN	5.0	0.0	2.0	0.0	0.0	0.0	17.0	88.0	1.0	0.0	0.0	23.0	0.0	0.0	0.0	0.0	3.0	4.0
NUM	0.0	0.0	0.0	0.0	0.0	0.0	8.0	0.0	99.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.0
PART	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	94.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PRON	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0
PROPN	4.0	0.0	2.0	0.0	0.0	0.0	0.0	8.0	0.0	0.0	0.0	55.0	0.0	0.0	0.0	0.0	2.0	21.0
PUNCT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	99.0	0.0	0.0	3.0	0.0	11.0
SCONJ	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	54.0	0.0	0.0	0.0	0.0
SYM	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	97.0	0.0	0.0
VERB	2.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	93.0	0.0
X	2.0	1.0	1.0	0.0	0.0	0.0	50.0	2.0	0.0	0.0	0.0	16.0	0.0	0.0	100.0	0.0	1.0	57.0
Total_Spacy	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

Figure 10: Frequency of Stanza UPOS corresponding to a Spacy UPOS. The columns represent Spacy tags and rows - Stanza's. On each intersection, you will find the percent of Spacy tokens from a given column interpreted as Stanza tokens of a given row.

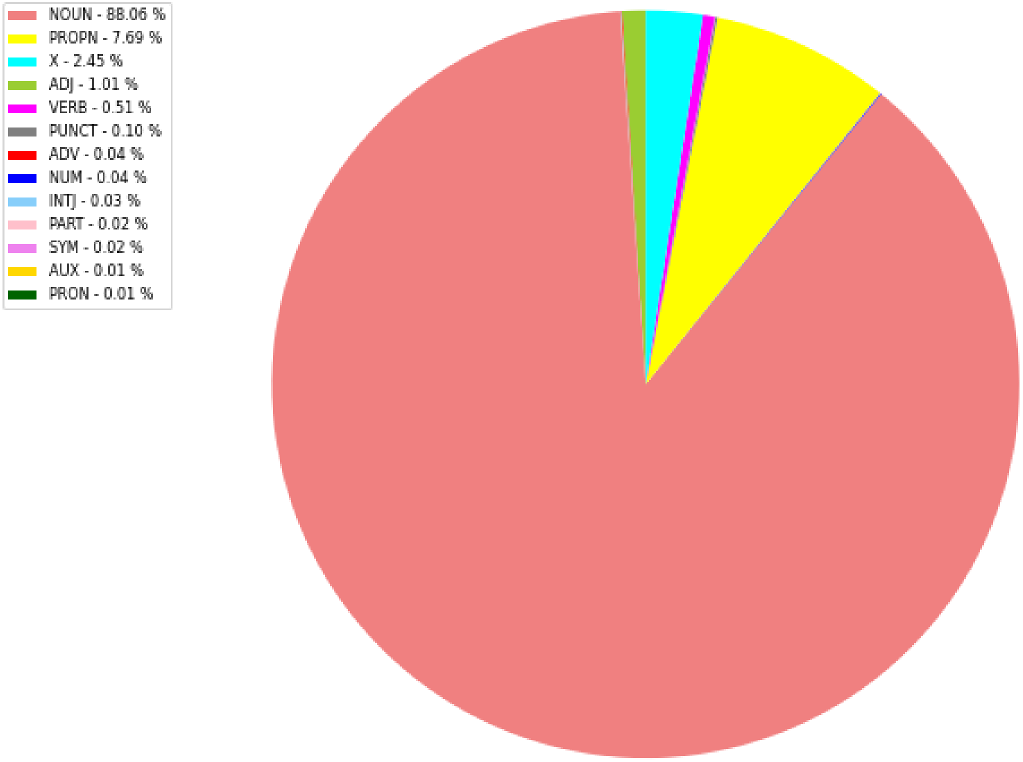


Figure 11: Stanza’s UPOS corresponding to Spacy’s NOUN.

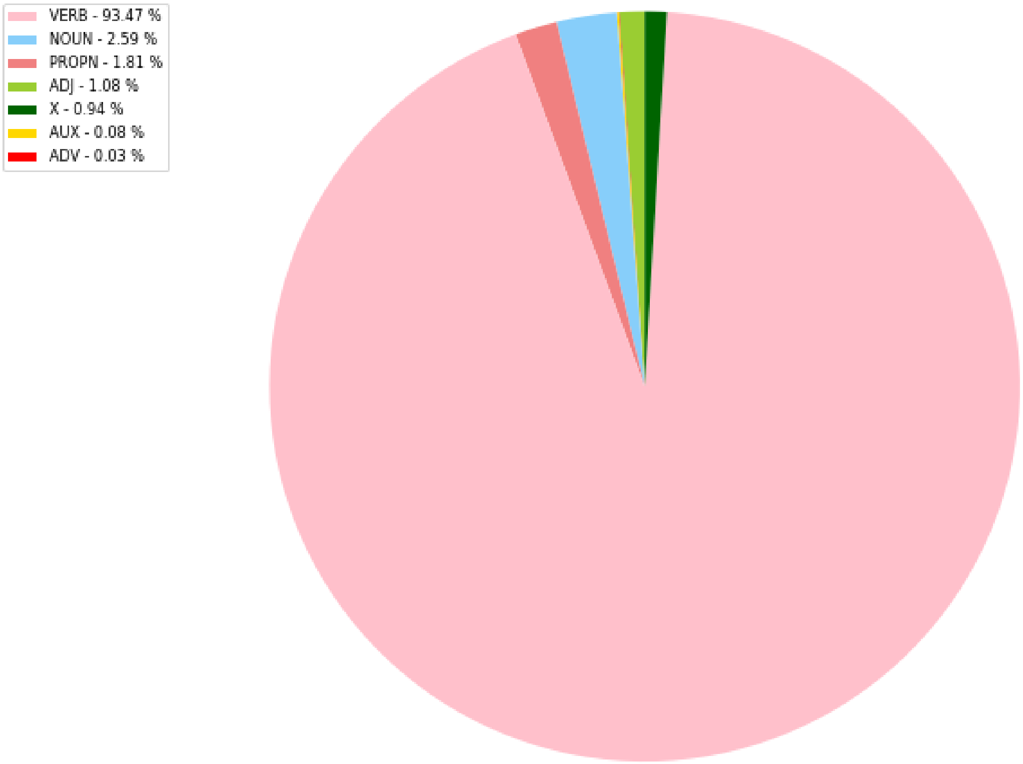


Figure 12: Stanza's UPOS corresponding to Spacy's VERB.

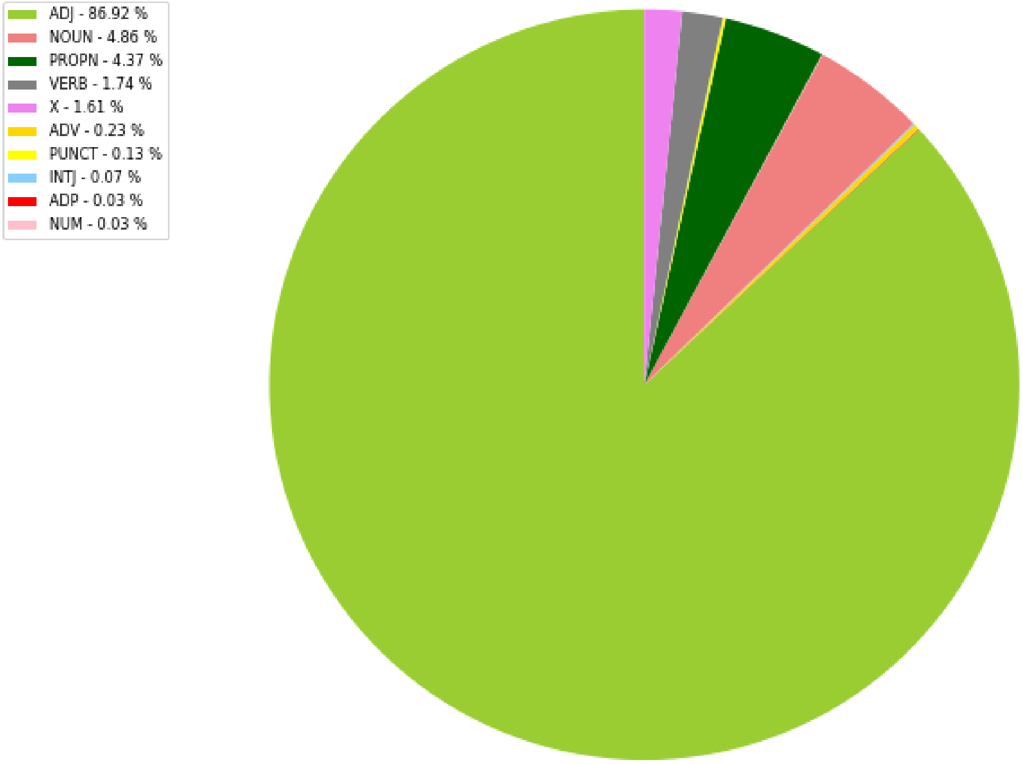


Figure 13: Stanza's UPOS corresponding to Spacy's ADJ.

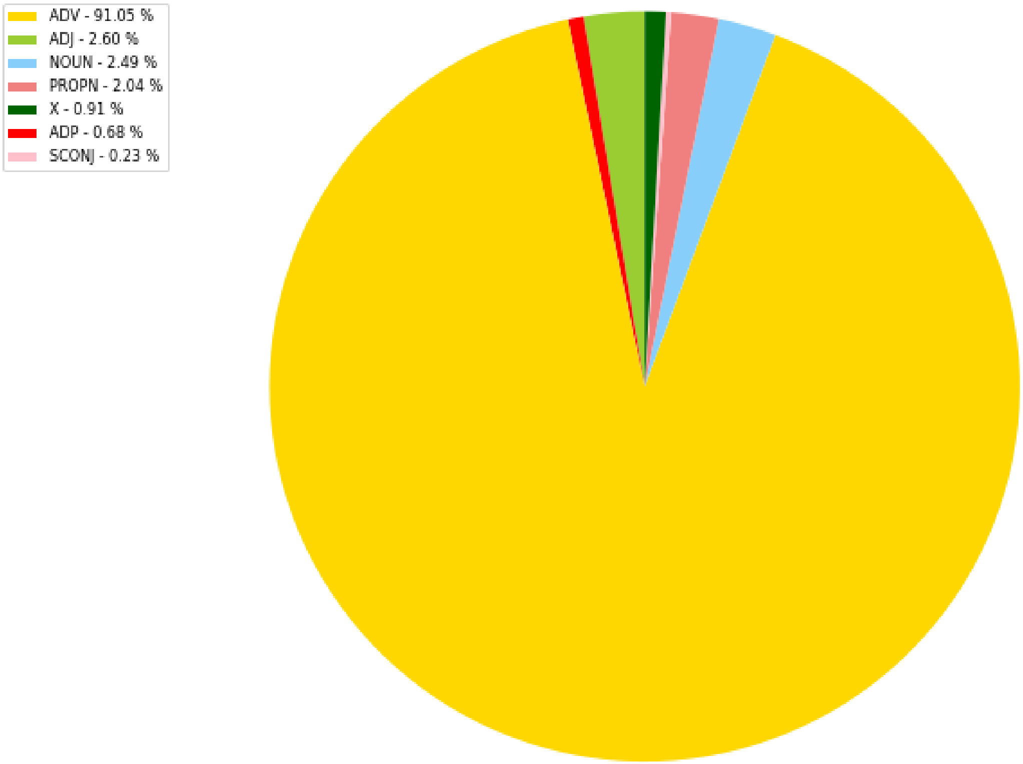


Figure 14: Stanza’s UPOS corresponding to Spacy’s ADV.