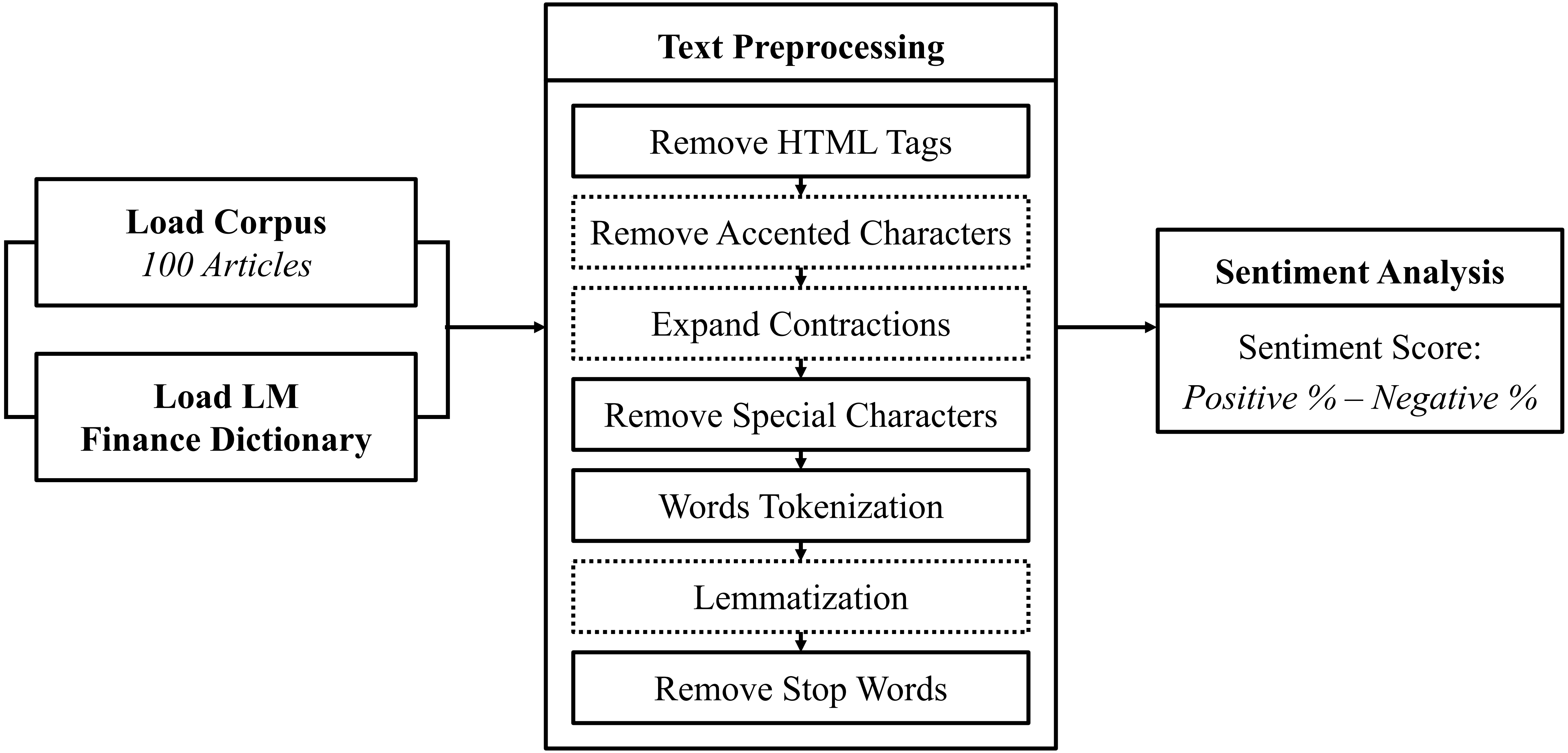
Sentiment Analysis for 100 Seeking Alpha Articles

Environment, Packages Required

- Environment: Python 3.7+ (Jupyter Notebook, PyCharm, etc.)
- Packages Required (contractions.py required)

Name	**Version**	**Summary**	**Home Page**
pandas	*0.24.2*	*Powerful data structures for data analysis, time series, and statistics.*	*http://pandas.pydata.org*
numpy	*1.18.0*	*NumPy is the fundamental package for array computing with Python.*	*https://www.numpy.org*
bs4	*0.0.1*	*Screen-scraping library.*	*https://pypi.python.org/pypi/beautifulsoup4*
pycontractions	*2.0.1*	*Intelligently expand and create contractions in text leveraging grammar checking and Word Mover's Distance.*	*https://github.com/ian-beaver/pycontractions*
numba	*0.44.1*	*Compiling Python code using LLVM.*	*http://numba.github.com*
nltk	*3.4.4*	*Natural Language Toolkit*	*http://nltk.org/*
spacy	*2.2.3*	*Industrial-strength Natural Language Processing (NLP) in Python*	*https://spacy.io*

Programming Pipelines



Rationale of Text Preprocessing

1. **Remove HTML Tags: YES**

   The raw corpus includes plenty of html tags.

2. **Remove Accented Characters: NO**

   Accented characters have insignificant impact or even no impact on sentiment analysis.

3. **Expand Contractions: NO**

   Expanding contractions will make the thing worse, in article #1067081, the wording *poor's* will be expanded as *poors*, and later, it cannot be identified as a negative word.

4. **Remove Special Characters: YES**

   The raw corpus includes plenty of special characters, such as & $ …, which will affect further preprocessing (tokenization) and sentiment analysis.

5. **Words Tokenization: YES**

   Tokenizing words is a necessary procedure before performing sentiment analysis.

6. **Words Lemmatization: No**

   The Loughran & McDonald finance dictionary itself already contains various forms corresponding to a word. Thus, performing lemmatization will lower accuracy!

Rationale of Text Preprocessing

7. **Words Stemming : No**

   As lemmatization and stemming have similar functions to a certain extent and in general, the former one will be superior than the later one in terms of conveying expressions. However, here the text preprocessing process will not perform either lemmatization or stemming.

8. **Remove Stop Words: YES**

   Different authors may have preferences in terms of using different stop words, with different frequency. To make sure that the sentiment score is comparable within the corpus, it would be better to strip the stop words. Note that positive words or negative words in the Lougrand & McDonald Finance Dictionary will be removed from the stop words list, here, the word "against".

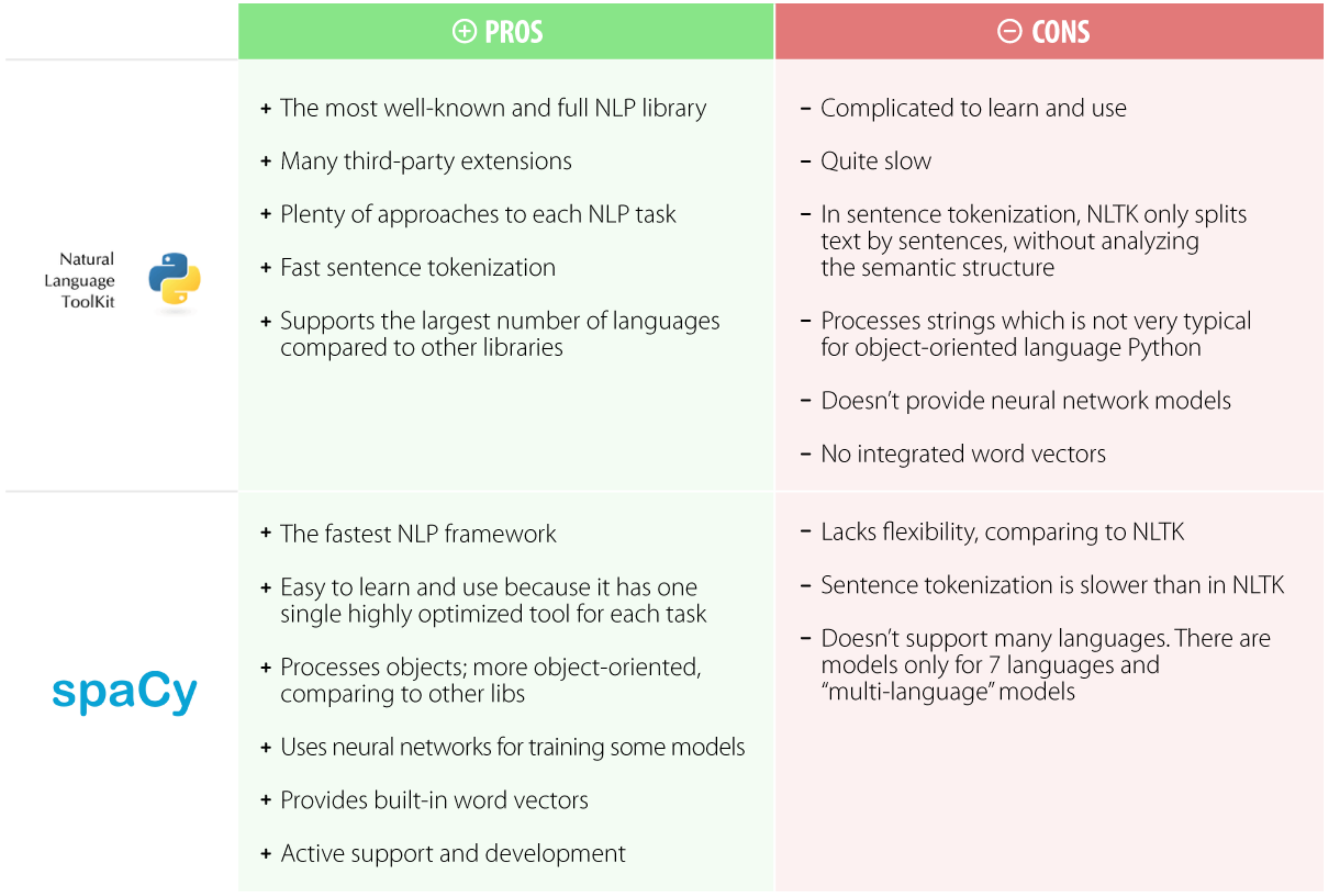
9. **Text Correction: NO**

   Considering that the 100 articles are posted on Seeking Alpha, a relatively professional and formal website, the phenomenon of spelling words wrong or using repeated characters to express specific feelings is in a low probability.

10. **Case Conversion: YES**

    As the words given in the Lougrand & McDonald Finance Dictionary are in the form of uppercase, making the form of the words in the corpus in line with the referenced sentiment dictionary is necessary.

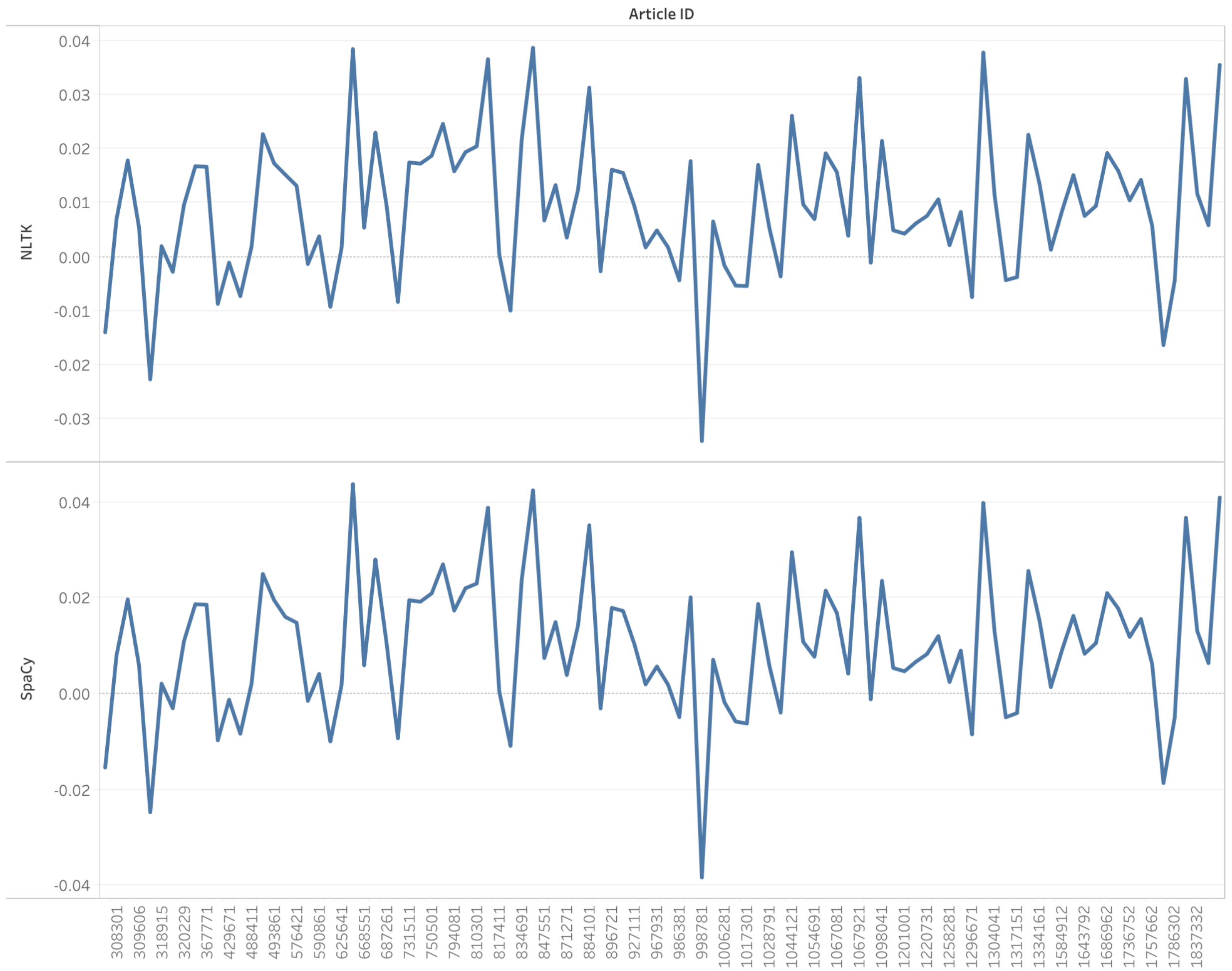
Comparison: NLTK, spaCy

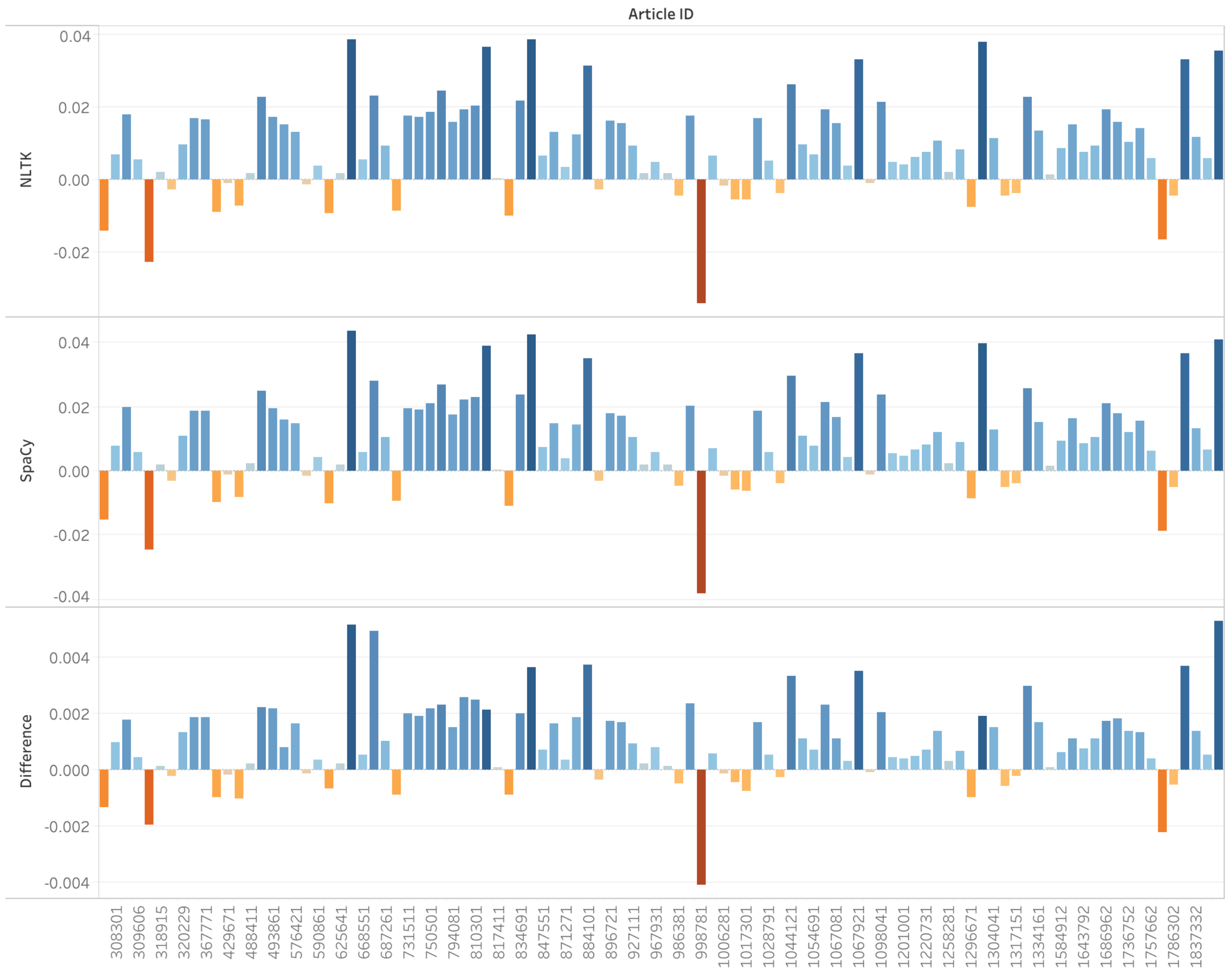


	⊕ PROS	⊖ CONS
Natural Language ToolKit	+ The most well-known and full NLP library + Many third-party extensions + Plenty of approaches to each NLP task + Fast sentence tokenization + Supports the largest number of languages compared to other libraries	– Complicated to learn and use – Quite slow – In sentence tokenization, NLTK only splits text by sentences, without analyzing the semantic structure – Processes strings which is not very typical for object-oriented language Python – Doesn’t provide neural network models – No integrated word vectors
spaCy	+ The fastest NLP framework + Easy to learn and use because it has one single highly optimized tool for each task + Processes objects; more object-oriented, comparing to other libs + Uses neural networks for training some models + Provides built-in word vectors + Active support and development	– Lacks flexibility, comparing to NLTK – Sentence tokenization is slower than in NLTK – Doesn’t support many languages. There are models only for 7 languages and “multi-language” models

References: https://activewizards.com/blog/comparison-of-python-nlp-libraries/

Comparison: NLTK, spaCy





Comparison: NLTK, spaCy

- The number of positive words and the number of negative words identified with *NLTK* are exactly the same as that identified with *spaCy*, respectively.
- As the built-in stop words list of *NLTK* contain 179 words while the built-in stop words list of *spaCy* contain 326 words, after removing stop words, there are more words left with *NLTK than with spaCy, i.e., total words-NLTK > total words-spaCy.*
- *Given that the sentiment score is the difference between the positive fraction and the negative fraction, text preprocessing with spaCy will somehow augment the sentiment, as shown in the figure. Specifically, if an article contain more positive words than negative words, the sentiment score will be positive and it will be higher with spaCy compared with NLTK. However, if an article contain more negative words than positive words, the sentiment score will be negative and it will be lower with spaCy compared with NLTK.*
- *77 out of 100 articles have positive sentiment scores, while the rest ones have negative sentiment scores.*