Benchmarking NLP Toolkits for Enterprise Application

Kok Weiying(✉), Duc Nghia Pham, Yasaman Eftekharypour, and Ang Jia Pheng

MIMOS Berhad, Kuala Lumpur, Malaysia
{kok.weiying, nghia.pham, yasaman.eftekhary, jp.ang}@mimos.my

Abstract. Natural Language Processing (NLP) is an important technology that motivates the form of AI applications today. Many NLP libraries are available for researchers and developers to perform standard NLP tasks (such as segmentation, tokenization, lemmatization, POS tagging, and NER) without the need to develop from scratch. However, there are some challenges in selecting the most suitable library such as data type, performance, and the compatibility. In this paper, we assessed five popular NLP libraries for performing the standard processing tasks on datasets crawled from different online news sources in Malaysia. The obtained results are analysed and differences of those libraries are listed. The goal of this study is to provide a clear view for users to select the suitable NLP library for their text analysis task.

Keywords: Natural language processing · Sentence segmentation · Tokenization · Lemmatization · POS tagging · Named entity recognition

1 Introduction

Natural Language Processing (NLP) plays an important role in current AI applications that require an understanding of human language such as contextual extraction, machine translation, content categorization, and so on. Some of the most common and practical examples of NLP-related applications are Google translate, Bing translate, spam email filtering, customer services, and voice assistants (e.g. Alexa, Cortana, Siri or Google Assistant).

Different libraries are available for researchers and developers to perform standard processing of widely-spoken languages, such as sentence segmentation, tokenization, part-of-speech tagging (POS), lemmatization and named entity recognition (NER). Factors such as update frequency, cost integration, language support, and accuracy performance need to be considered for implementing a robust application [1].

In this paper, we will focus on the comparison between five popular NLP libraries that are publicly available (i.e. CoreNLP, NLTK, OpenNLP, SparkNLP, and spaCy) to help in sentence segmentation, tokenization, lemmatization, POS tagging, and NER tasks. In summary, the selected libraries are reviewed and evaluated based on the choice of programming language, license type, supported NLP tasks, and the algorithms used. The results found using the pre-trained models of these NLP libraries are compare in detailed.

A. C. Nayak and A. Sharma (Eds.): PRICAI 2019, LNAI 11672, pp. 289–294, 2019.
https://doi.org/10.1007/978-3-030-29894-4_24

2 Common NLP Tasks

Table 1 lists the common NLP tasks and their dependencies/difficulties in processing.

Table 1. List of common NLP tasks.

NLP Task	Description	Difficulties/Dependencies
Sentence segmentation	Divide a bunch of text into sentences	- Language dependency: each written language has its own sentence structure or rules such as the full stop punctuation used to end a sentence for Chinese (。) and English (.) is in a different form and has a different meaning in a context [2] - Application dependency: there is no absolute definition on what constitutes a sentence and is relatively arbitrary distinction across different written languages [2] - Corpus dependency: a robust NLP approach is needed with the increasing number of text corpora that contain irregular features, punctuation or misspelling that are unable to be processed by algorithm that is trained to process well-formed sentences [2]
Tokenization	Break a sentence into tokens (words, numbers or punctuation)	Difficulties in tokenization: - Space-delimited languages (Latin alphabet): Tokenization ambiguity exists with the uses of punctuation such as apostrophes, hyphen, commas, etc. - Unsegmented languages (Chinese, Japanese or Thai) do not contain word boundaries or whitespace between each word where additional lexical and morphological information is needed while tokenizing these languages [2]
Lemmatization	Remove the inflectional ending of a word to lemma	Lemmatization provides a better precision are usually used to improve the performance of text similarity metrics [3]
POS tagging	Classify words in a sentence to the proper morphosyntactic tags	The most common POS tagger for English is the Penn Treebank tag set which contains 36 POS tags and 12 other tags [4]
Named entity recognition	Identify unique entities and classify them into predefined categories (e.g. person, location, organization, etc.)	Linguistic grammar-based techniques show a higher precision but lower recall and time consuming for expert linguist to craft the rules whereas statistical machine learning models required large amount of annotated training data [5]. Both methods suffer from shortcomings on the maintenance and development of large scale NER system [5]

Table 2. List of five popular NLP libraries. They all provide pre-trained models for the 5 common NLP tasks in Table 1, except that OpenNLP doesn't support lemmatization.

Library	Description	Licence	Language
Stanford CoreNLP	Highly flexible and extensible. Can be used as an integrated toolkit with a wide range of grammatical analysis tools and provides a number of wrappers that can be used in various major modern programming languages [6]	GPL v3	Java
NLTK	Provides ready-to-use computational linguistics courseware. Contains over 50 corpora and lexical sources such as Penn Treebank Corpus, Open Multilingual Wordnet and a suite of text processing libraries for almost all NLP [7]	Apache v2.0	Python
OpenNLP	Contains various components that enable user to build a full NLP pipeline to execute respective NLP tasks, or train and evaluate a model via its API [8]	Apache v2.0	Java
SparkNLP	A natural language processing library built on top of Apache Spark ML. SparkNLP provides simple, performance & accurate NLP annotations for machine learning pipelines that can be scale easily in a distributed environment [9]	Apache v2.0	Python
spaCy	Designed specifically for production use which helps to build applications that process a large volume of text [10]. spaCy can be used to build information extraction or natural language understanding system or pre-processing text for deep learning [10]	MIT	Python

3 NLP Libraries

In this study, five NLP libraries (as shown in Table 2) are selected based on (i) the availability of pre-trained model, (ii) the ability to perform the five common NLP tasks mentioned above, (iii) the ability to process English language text, and (iv) the support of Java or Python programming language.

Table 3. The accuracy of 5 NLP libraries for sentence segmentation, tokenization, lemmatization & POS tagging based on the annotated data.

Library	#Sentences	#Tokens	Segmentation (%)	Token. (%)	Lemma. (%)	POS (%)	NER (%)
CoreNLP	117	881	**96.85**	**99.89**	**97.26**	97.17	**97.67**
NLTK	111	871	**96.85**	96.69	82.67	92.45	94.43
OpenNLP	116	870	90.55	99.09	N/A	96.89	96.72
SparkNLP	150	881	74.16	98.97	96.01	93.56	93.08
spaCy	150	906	75.59	98.86	90.08	**97.20**	93.92

Table 4. Detailed comparison of results processed by 5 NLP libraries and human. Results that different to human annotation are bold.

Task	Human	CoreNLP	NLTK	OpenNLP	SparkNLP	spaCy
Tokenization	“anti-graft”	“anti-graft”	“anti-graft”	“anti-graft”	“anti-graft”	**“anti”, “-”, “graft”**
	“KG-DWN-98/2”	“KG-DWN-98/2”	“KG-DWN-98/2”	“KG-DWN-98/2”	“KG-DWN-98/2”	**“KG”, “-”, “DWN-98/2”**
	“US$8.5mil”	**“US$”, “8.5”, “mil”**	**“US”, “$”, “8.5mil”**	**“US$”, “8.5mil”**	“US$8.5mil”	**“US$”, “8.5mil”**
Lemmatization	“was”	“was”	**“wa”**	“was”	“was”	“was”
	“as”	“as”	**“a”**	“as”	“as”	“as”
	“MyEG Services Bhd”	“MyEG Services Bhd”	“MyEG Services Bhd”	“MyEG Services Bhd”	“MyEG Services Bhd”	**“myeg services bhd”**
POS tagging	“co” (NN), “-” (HYPH), “founder” (NN)	**“co-founder” (NN)**	“co” (NN), “-” (HYPH), “founder” (NN)	“co” (NN), “-” (HYPH), “founder” (NN)	“co” (NN), “-” (HYPH), “founder” (NN)	“co” (NN), “-” (HYPH), “founder” (NN)
NER	“AirAsia” (ORG)	**“AirAsia” (LOC)**	**“AirAsia” (PER)**	**“AirAsia” (PER)**	**“AirAsia” (PER)**	**“AirAsia” (LOC)**

4 Results and Discussion

We collected 171 news articles from Malaysian news sites from July to August 2018. Images and unwanted symbols were removed. We then ranked these articles based on the number of words, named entities, different sentence structure and punctuation used. Finally, ten highest ranking articles were manually annotated and processed with sentence segmentation, tokenization, lemmatization, POS tagging and NER.

Table 3 shows the results of these 5 NLP libraries on the 10 highest ranking news articles. The results show that CoreNLP has the highest accuracy in four of NLP tasks (segmentation, tokenization, lemmatization, and NER) and slightly (0.03%) worse than spaCy on POS tagging. Table 4 highlights the differences between human annotated results and these 5 libraries on processing the 5 common NLP tasks.

The available pre-trained models are unable to detect Malaysian named entities: they were either left untagged or incorrectly tagged. Hence, we selected CoreNLP, OpenNLP and, spaCy – the three best libraries on the other four NLP tasks – and retrained their NER models using our local news dataset of 171 articles (80% training and 20% testing). Table 5 shows the results of these three libraries on NER tagging. CoreNLP and spaCy both reached an F-score of 0.78 whilst OpenNLP only scored 0.62.

Table 5. Precision, Recall, F-score results for NER of CoreNLP, OpenNLP, and spaCy.

Library	Algorithm	Tagging format	Precision	Recall	F-score
CoreNLP	Conditional Random Field [11]	where labeles are separated by a tab "\t" e.g. "word \tLABEL"	0.83	0.73	0.78
OpenNLP	Maximum Entropy [8]	each sentence has a mark with entity e.g. "<START: person> Entity <END>"	0.87	0.48	0.62
spaCy	Word embedding strategy using sub-word features and "Bloom" embedding, CNN with residual connections, transition-based approach to named entity parsing [10]	Wikipedia scheme IOB Scheme BILUO Scheme	0.79	0.77	0.78

5 Conclusion

Selection of the right NLP library is critical in developing an NLP-based application as it affects the accuracy of analysis tasks. Our results showed that both CoreNLP and spaCy produced higher accuracy than others. Between the two libraries, spaCy is significantly faster than CoreNLP, up to 10 times faster on certain tasks. We hope that our findings can help developers or researchers in selecting the right NLP library for their tasks, saving them the time and effort to retrain and compare different libraries for common NLP tasks.

References

1. Al Omran, F.N.A., Treude, C.: Choosing an NLP library for analyzing software documentation: a systematic literature review and a series of experiments, pp. 187–197. IEEE Press, Piscataway (2017)
2. Palmar, D.D.: Text preprocessing. In: Indurkhya, N., Damerau, F.J. (eds.) Handbook of natural language processing, vol. 2, pp. 9–30. CRC Press, Boca Raton (2010)
3. Aker, A., Petrak, J., Sabbah, F.: An extensible multilingual open source lemmatizer. In: Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP, pp. 40– 45. ACL (2017)
4. Marcus, M., Beatrice, S., Mary, A.: Building a large annotated corpus of. English: The Penn Treebank (1993)
5. Epaminondas, K., Tatar, D., Sacarea, C.: Named entity recognition. In: Natural Language Processing: Semantic Aspects, pp. 297–309. CRC Press, Boca Raton (2013)
6. Pinto, A., Gonçalo Oliveira, H., Oliveira Alves, A.: Comparing the performance of different NLP toolkits in formal and social media Text. In: 5th Symposium on Languages, Applications and Technologies (SLATE2016). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2016)

7. Loper, E., Bird, S.: NLTK: the natural language. arXiv preprint cs/0205028 (2002)
8. Foundation, T.A.: Apache OpenNLP Developer Documentation. (The Apache Software Foundation) (2011). https://opennlp.apache.org/docs/1.9.1/manual/opennlp.html, Accessed 01 Mar 2019
9. John Snow Labs: SparkNLP - Documentation and Reference (2019). https://nlp.johnsnowlabs.com/components.html, Accessed 11 Mar 2019
10. Honnibal, M.: Introducing spaCy (2016). https://explosion.ai/blog/introducing-spacy, Accessed 01 Mar 2019
11. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55–60 (2014)