A Comparative Study of Stanford NLP and Apache Open NLP in the view of POS Tagging

Jay Nanavati, Yogesh Ghodasara

Abstract- To perform a comparative study of two popular Natural Language Processing tools – Stanford NLPand Apache Open NLP, is the main objective of this paper. This paper also provides an insight into use of these two tools for analysis of requirements specification expressed in Natural Language English.

Keywords- NLP, Requirements Specifications, Part Of Speech (POS) tagging, Tokens, Tag set

I. INTRODUCTION

Requirement analysis phase translates the ideas in the minds of the clients into a formal document. The requirements are critically analyzed and then abstraction of it, is created. This is known as requirement model. Natural language is quite informal in nature, especially, when it is used for elicitation of system requirements. English is also inconsistent as majority of English words have multiple senses and a single sense can be reflected by multiple words in English. A variety of styles have been introduced by people using English as far as writing and speaking are concerned. This has become more complicated with technology-related terms' acceptance in the standard dictionaries. English language has a vocabulary of approximately 600000 words, built over thousand years.[1] All these facts have an adverse effect on the first and foremost important step of SDLC i.e. Requirements Analysis. Many CASE tools have been developed for the automation of one or more phases of software development. Most of these tools use one or other Natural Language Processing toolkit. [2, 3] We have selected two such tools for a comparative study of their performance in the view of requirements analysis.

II. PART-OF-SPEECH (POS) TAGGING

It is a process of assigning part-of-speech tags to the tokens (i.e. words) in a corpus.The phrase “Part-of-speech” collectively identify noun, verb, adjective, preposition and adverb present within the text. POS tagging is useful in information retrieval. It is also useful for Text-to-Speech translation and Word Sense Disambiguation. [4] A fine-grained tag set is used to assign tags to words in the text. Penn Treebank tag set is an example of fine-grained tag set in which 36 tags are there. [5]

III. STANFORD LOG-LINEAR PART-OF-SPEECH TAGGER

This software is a Java implementation of the log-linear part-of-speech taggers. The basic download contains two trained tagger models for English. The full download contains three trained English tagger models, an Arabic tagger model, a Chinese tagger model, a French tagger model, and a German tagger model. Both versions include the same source and other required files. The tagger can be retrained on any language, given POS-annotated training text for the language. [6]

IV. APACHE OPEN NLP

The Open NLP POS Tagger uses a probability model to predict the correct pos tag out of the tag set. To limit the possible tags for a token a tag dictionary can be used which increases the tagging and runtime performance of the tagger. [7]

A. *The Experiment*

We have primarily considered sentences with following tenses and corresponding forms:

Tense	Forms
Present	Simple Present Tense Continuous Present Tense Perfect Present Tense
Past	Simple Past Tense Continuous Past Tense Perfect Past Tense
Future	Simple Future Tense Continuous Future Tense Perfect Future Tense

Thus, we mainly have sentences with 9 different tense.
We have further expanded our sentence base by considering the following forms in which the statements may occur:

Sr. No.	Form
1	Single verb
2	Multiple verbs
3	Verb similar to noun
4	Negative
5	Emphasis
6	Direct/Indirect speech
7	Highly Complex

Finally, we have prepared and used a sentence base of 40 different sentences. This serves as input to the actual POS tagger programs.

B. *Programs*

We have used two programs: one uses Stanford POS tagger and the other uses Apache OpenNLP POS tagger. As we have mentioned earlier, our objective is compare performance offered by these two POS taggers. Both the programs are implements in Java programming language.

C. *Issues*

The basic set of tags can be easily established for any language but the correct and/or perfect set of tags may not be designed. This is so because each language including English suffers from ambiguity. There are many words in English which represent verb, noun and adjective. Live, attempt, dry,

Revised Version Manuscript Received on November 05, 2015.

Jay Nanavati, Research Scholar, School of Science, RK University, Rajkot, (Gujarat) India.
Dr. Yogesh Ghodasara, Associate Professor, College of Agricultural I.T., Anand Agriculture University, Anand, (Gujarat) India.

kick are just a few simple examples of such words.

With new words being added to the standard dictionary every year, it becomes imperative to update the dictionary which is used for POS tagging. Consider these new arrivals: selfie, tweet, for example. Words which are originally from other languages and used in English also pose a great challenge for POS tagging. Consider these words: prima facie, status quo, habeas corpus, ad hoc. Words which are composite in nature that is made up of two more words also contribute to confusion. Consider the sentence: Look that word up in the dictionary. Here “look" and "up" are used as a single verbal unit, despite the possibility of other words coming between them. Penn and many other tagsets break hyphenated words, possessives and contractions into separate tokens before finally POS tagging them. However, it has not proven as a complete solution to the problem. A few verbs occurs in quite different grammatical contexts, Complicating the issue. [8]

Results and Analysis

Table: 1 Stanford NLP vs. Apache Open NLP – Comparison of POS tagging accuracy

Sr. No.	Type of Sentence	No. of Sentences tested	No. of tokens tagged	Performance (No. of tokens correctly tagged)		Accuracy (%)	
				Stanford NLP	Apache Open NLP	Stanford NLP	Apache Open NLP
1	Simple Present Tense	5	20	20	20	100	100
2	Continuous Present Tense	5	20	20	20	100	100
3	Perfect Present Tense	5	20	20	20	100	100
4	Simple Past Tense	5	20	20	20	100	100
5	Continuous Past Tense	5	20	20	20	100	100
6	Perfect Past Tense	5	20	20	20	100	100
7	Simple Future Tense	5	20	20	20	100	100
8	Continuous Future Tense	5	20	20	20	100	100
9	Perfect Future Tense	5	20	20	20	100	100
10	Ambiguous (Verb similar to Noun)	10	50	44	42	88	84
11	Use of conjunctives	5	25	24	23	96	92
12	Negative	5	25	25	25	100	100
13	Emphasis	5	25	21	20	84	80
14	Direct speech	10	50	47	45	94	90
15	Indirect speech	10	50	47	44	94	88
16	Highly Complex	20	200	171	167	86	84
Total		110	605	559	546	92	90

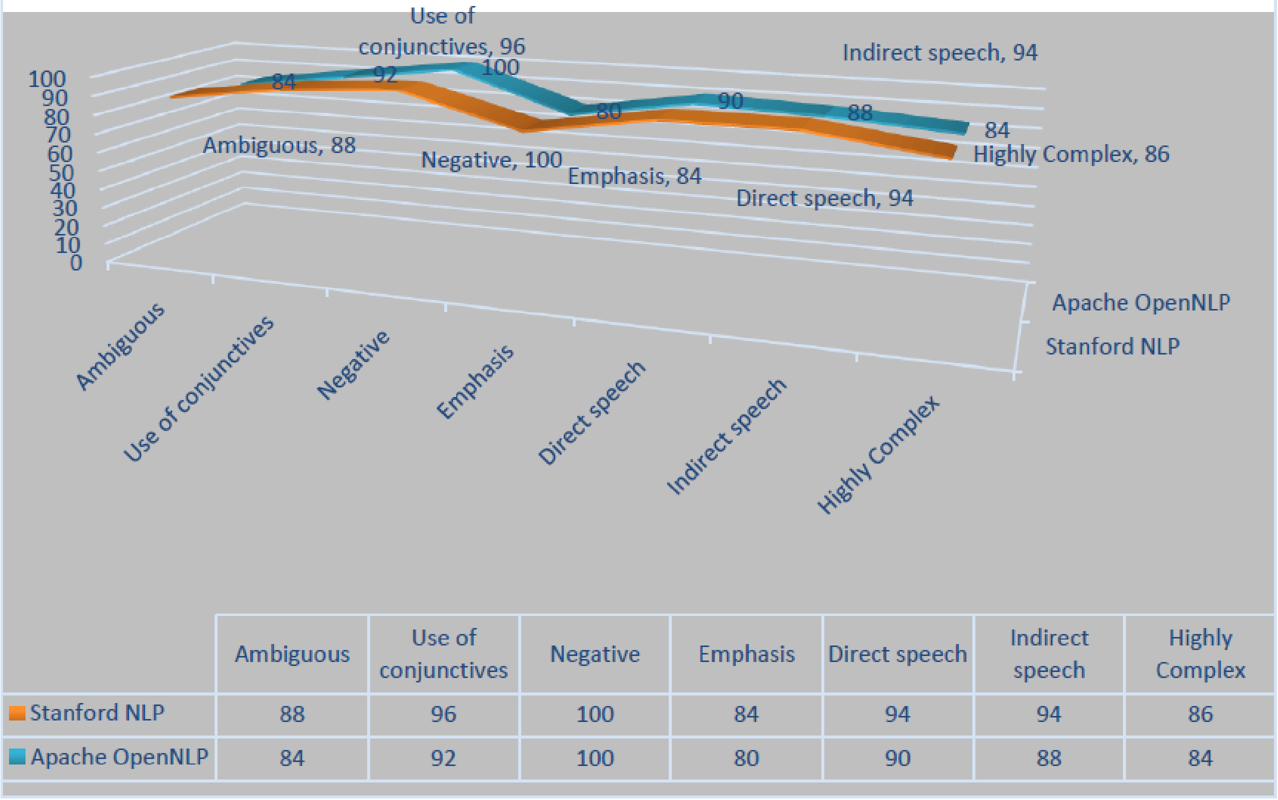

Figure: 1Stanford NLP vs. Apache Open NLP in critical sentence forms

Table: 2 Stanford NLP vs. Apache Open NLP – Comparison of time taken for POS tagging

Sr. No.	Type of Sentence	No. of Sentences tested	No. of tokens tagged	Performance Time taken (sec)		Difference in time taken (sec)
				Stanford NLP	Apache Open NLP	
1	Simple Present Tense	5	20	0.833	1.366	0.533
2	Continuous Present Tense	5	20	0.840	1.380	0.54
3	Perfect Present Tense	5	20	0.849	1.383	0.534
4	Simple Past Tense	5	20	0.825	1.300	0.475
5	Continuous Past Tense	5	20	0.800	1.350	0.55
6	Perfect Past Tense	5	20	0.798	1.275	0.477
7	Simple Future Tense	5	20	0.777	1.195	0.418
8	Continuous Future Tense	5	20	0.801	1.297	0.496
9	Perfect Future Tense	5	20	0.775	1.200	0.425
10	Ambiguous (Verb similar to Noun)	10	50	1.230	1.760	0.53
11	Use of conjunctives	5	25	0.988	1.036	0.048
12	Negative	5	25	0.950	1.000	0.05
13	Emphasis	5	25	1.610	2.680	1.07
14	Direct speech	10	50	1.575	2.090	0.515
15	Indirect speech	10	50	1.644	2.064	0.42
16	Highly Complex	20	200	3.210	4.205	0.995
Total		110	605			

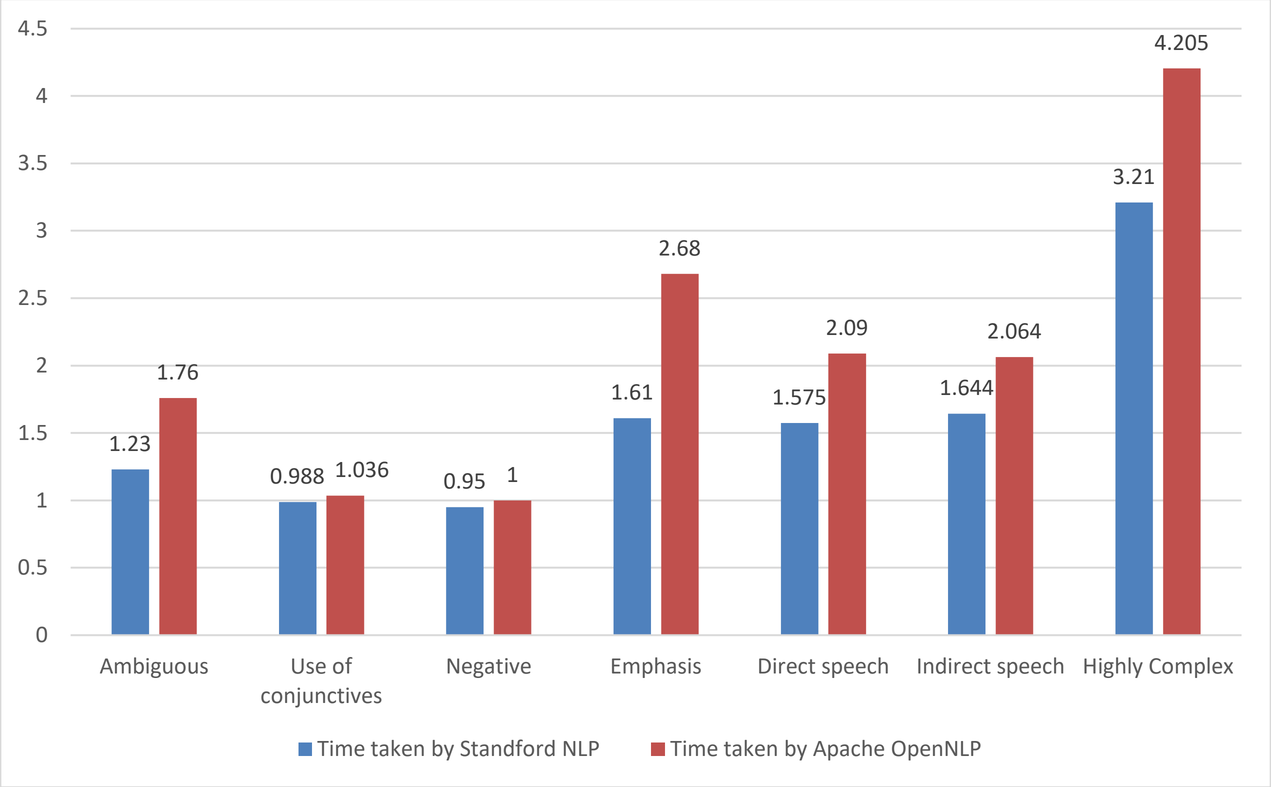

Figure: 2 Stanford NLP vs. Apache Open NLP - Comparison of time taken for POS tagging

Table: 3 Stanford NLP vs. Apache Open NLP – % Difference in time taken for POS tagging

Category	Time taken by Standford NLP (Ts) sec	Time taken by Apache Open NLP (Ta) sec	Difference (Ta – Ts) sec	% Difference w.r.to Ts
Ambiguous	1.23	1.76	0.53	43
Use of conjunctives	0.988	1.036	0.048	5
Negative	0.95	1	0.05	5
Emphasis	1.61	2.68	1.07	66
Direct speech	1.575	2.09	0.515	33
Indirect speech	1.644	2.064	0.42	26
Highly Complex	3.21	4.205	0.995	31

It can be deduced from Table-1 that for simple sentences (i.e. Sr. No. 1 to 9), which are free from any kind of ambiguity, speech and conjunctives, accuracy of both tools is 100% and there is no difference in their performance in terms of no. of tokens correctly tagged. As the input sentences become complicated, Stanford NLP is found to be more correct as compared to Apache Open NLP. Further, as shown in Figure-1, the difference in select areas is also not uniform. Table-2 describes performance comparison in terms of time taken by these two softwares to complete POS tagging of input. A careful observation reveals that Apache Open NLP takes more time than Stanford NLP takes in all the cases. Study of Table-3 shows % difference in time with respect to Ts. It can be estimated that Apche Open NLP consumes 29% more time than Stanfor NLP.

V. CONCLUSION

Although the accuracy of POS tagging of these two softwares is fairly comparable, it is evident that Stanford NLP takes less time than Apache Open NLP. Citing the observations mentioned above it can be concluded that Stanford NLP is better than Apache Open NLP as long as POS tagging is concerned.

REFERENCES

1. http://www.oed.com (Dt. 3/6/15)
2. N. Boyd, “Using Natural Language in Software Development”, Journal of Object Oriented Programming, Feb. 1999.
3. M. Osborne, C.K. MacNish, "Processing Natural Language Software Requirement Specifications", Proceedings of the 2th International Conference on Requirements Engineering, IEEE, 15-18 April 1996, pp. 229-236
4. http://www.cs.umd.edu/~nau/cmsc421/part-of-speech-tagging.pdf (Dt. 3/6/15)
5. https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html (Dt. 6/6/15)
6. Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252-259.
7. https://opennlp.apache.org/documentation/1.5.3/manual/opennlp.html#tools.postagger (Dt. 12/6/15)
8. https://en.wikipedia.org/wiki/Part-of speech_tagging#Unsupervised_taggers (Dt. 12/6/15)