Stanford CoreNLP Performance and Accuracy in Comparative Perspective

Comparative analysis of NLP packages

Analyses based on tokenization and POS tagging

Al Omran and Treude (2017) carried out a systematic analysis of some 232 software engineering conference papers between 2011 and 2015 that contained the words “natural language.” Only 33 papers mentioned the specific library/package used. Of those, the majority use the Stanford CoreNLP library.” “But “not a single paper … included a thorough justification… of… their choice of NLP library.”

Their analyses are based on **tokenization and POS tagging for different collections of text data**.

Al Omran and Treude (2017) conclude: “the choice of the best NLP library depends on the task and the source.” “spaCy provides the best overall performance on the data that we used in our experiments,” “the best choice of an NLP library depends on which part of the NLP pipeline [i.e., which specific tool] is going to be employed.”

Analyses based on parsers

In a comparative analysis of different parsers, Choi et al (2015) argue that different parsers achieve different levels of accuracy depending upon sentence well formedness, average sentence length and standard deviation, and different types of "noisy text." pipeline

in their comparative analysis of different NLP packages, argue that different NLP packages perform differently for different NLP tasks depending upon the corpus to be analyzed.

spaCy is substantially faster than many other libraries; a conclusion also supported by Choi et al. 2015.

Why Stanford CoreNLP

In a systematic analysis of some 232 software engineering conference papers between 2011 and 2015 that contained the words "natural language," Al Omran and Treude (2017) found that only 33 papers mentioned the specific library/package used. Of those, the majority use the Stanford CoreNLP library." "But "not a single paper … included a thorough justification… of… their choice of NLP library." We chose the Stanford CoreNLP package for the following reasons.

1. Stanford CoreNLP fares well for its popularity among NLP packages (Al Omran and Treude 2017), albeit with no justification.
2. It is *always* included as one of the contenders in any comparative analysis of specific NLP tools with overall comparable performance (POS tagger: Xue and Zhang 2020; NER tagger: Atdag and Labatut 2013; Dlugolinsky et al. 2013; Reshmi and Balakrishnan 2018; Dawar et al. 2019; Shelar et al. 2020; parser: Choi et al. 2015; coreference resolution: Beheshti et al. 2017; NLP packages: Al Omran and Treude 2017).
3. Stanford CoreNLP NER for Person, Location, and Organization give comparatively better performance even with poor-quality OCRed texts (e.g., Rodriquez et al. 2012; Ruokolainen and Kettunen 2020).
4. Stanford CoreNLP also offers very useful tools beyond the basic lemmatizing, POS, NER, and basic parser, such as OpenIE relation extractor or the sentiment analysis, coreference resolution, gender, and quote annotators.

CoreNLP POS tagger accuracy

In a comparative analysis of seven POS taggers, Xue and Zhang (2020) found that Stanford CoreNLP outperformed all others in their experiment. Notwithstanding, the Stanford CoreNLP POS tagger's performance has been shown to vary with the type of corpus analyzed. It achieves close to **97%** accuracy on *Wall Street Journal* data (Toutanova et al. 2003). Accuracy drops to **85.85%** on *Twitter* data (Gimpel et al. 2011) and to **74.3%** accuracy on a variety of texts that use *African American Vernacular English* (Jørgensen et al. 2016).

CoreNLP NER (Named Entity Recognition) accuracy

Accurate identification of NER (Named Entity Recognition) tags have become crucial in several downstream applications. Geocoding, for instance, is based on accurate NER tags for locations.

The identification of persons and organizations requires accurate NER tags for Knowledge graphs systems (e.g., DBpedia or YAGO). Similarly for numbers, such as telephone numbers.

Several comparative analyses of Stanford CoreNLP NER performance put the tool among the top performers (Rodriquez et al. 2012; Atdağ and Labatut 2013; Mishra et al. 2020). But whichever tool one uses, accuracy deteriorates with unclean names (e.g., non-capitalized Persons, Organizations or Locations; Mishra et al. 2020)

CoreNLP parser accuracy and performance

In a comparative analysis of the performance of ten different dependency parsers in terms of speed and accuracy, Choi et al. (2015) show that the Stanford CoreNLP's is well within the narrow range of all other packages, ultimately depending upon specific tool and corpus.

CoreNLP coreference resolution

Comparative analyzes of coreference resolution approaches show that the Stanford CoreNLP coreference resolution algorithm performs among the best (e.g., Beheshti et al. 2017).

New kids on the block

spaCy (2015)

spaCy was first released in 2015; a package developed Matthew Honnibal in Python and Cython. In all comparative analyses, spaCY is ranked as a formidable NLP newcomer, particularly for speed (Choi et al. 2015; Al Omran and Treude 2017; Qi et al. 2020). In Choi et al.'s experiments (2015), spaCy parses 755 sentence per seconds compared to ClearNLP 555 and CoreNLP 465; 13,963 tokens per seconds (10,271 and 8,602). But spaCy outperforms competitors as an NER tagger (Shelar et al. 2020) and parser (Choi et al. 2015).

Stanza (2019/2020)

Stanza is a new NLP Python package released by the Stanford NLP Group. Released in 2019 with the name StanfordNLP, it took the name Stanza a year later (Qi et al. 2020). Compared to the 7 languages that the Stanford CoreNLP Java version supports, Stanza supports 66 different languages. In Qi et al.'s performance comparison (2020) with spaCy and FLAIR, Stana outperforms both in accuracy but, as the authors acknowledge "Stanza's extensive use of accurate neural models makes it take significantly longer than spaCy to annotate text."

Issues affecting accuracy

The following issues have been shown to affect performance and accuracy.

1. ***Training corpora*** (e.g., Stanford CoreNLP *Wall Street Journal* vs. NLTK wider training data);

2. ***Sentence length*** **(**for all packages, performance and accuracy decline rapidly with longer sentences; Choi et al. 2015);
3. ***Sentence well formedness*;**
4. *Noisy text*

References

Al Omran, F., A. Nasser, and C. Treude. 2017. "Choosing an NLP library for analyzing software documentation: a systematic literature review and a series of experiments." *Proceedings of the 2017 IEEE/ACM 14th International Conference on Mining Software Repositories*, 2017 / pp.187-197.

Atdag, S. and V. Labatut. 2013. "A comparison of named entity recognition tools applied to biographical texts." *2nd International Conference on Systems and Computer Science (ICSCS)* 228–33. doi:10.1109/IcConSCS.2013.6632052

Beheshti, S.-M.-R., B. Benatallah, S. Venugopal, S. H. Ryu, H. R. Motahari-Nezhad, and W. Wang. 2017. "A systematic review and comparative analysis of cross-document coreference resolution methods and tools." *Computing*, 99:313–349 DOI 10.1007/s00607-016-0490-0

Choi, J. D., J. Tetreault, and A. Stent. 2015. "It Depends: Dependency Parser Comparison Using a Web-based Evaluation Tool." *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 387–396, Beijing, China, July 26-31, 2015.

Dawar, K., A. J. Samuel, and R. Alvarado. 2019. "Comparing topic modeling and named entity recognition techniques for the semantic indexing of a landscape architecture textbook." *Systems and Information Engineering Design Symposium (SIEDS)*, Charlottesville, VA, IEEE.

Dlugolinsky, S., M. Ciglan, and M. Laclav ́ık. 2013. "Evaluation of named entity recognition tools on microposts." *IEEE 17th International Conference on Intelligent Engineering Systems* 197–202. doi: 10.1109/INES.2013.6632810.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J. and Smith, N. A. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers, vol. 2. Association for Computational Linguistics*, pp. 42–7.

Jørgensen, A., Hovy, D. and Søgaard, A. 2016. "Learning a POS tagger for AAVE-like language." In *Proceedings of NAACL-HLT*. San Diego, CA, USA, pp. 1115–20.

Mishra*,* S.*,* S. He, and L. Belli. 2020. "Assessing Demographic Bias in Named Entity Recognition." *arXiv*:2008.03415v1 [cs.CL] 8 Aug 2020.

Qi, P., Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning. 2020. "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages." *arXiv*:2003.07082v2 [cs.CL] 23 Apr 2020.

Reshmi, S., and K. Balakrishnan. 2018. Enhancing Inquisitiveness of chatbots through NER integration. International Conference on Data Science and Engineering (ICDSE). doi: 10.1109/ICDSE.2018.8527788.

Rodriquez, K. J., M. Bryant, T. Blanke, and M. Luszczynska. 2012. Comparison of named entity recognition tools for raw OCR text. Proceedings of KONVENS 410–14. doi:

10.13140/2.1.2850.3045Shelar, H., G. Kaur, N. Heda, and P. Agrawal. 2020. “Named Entity Recognition Approaches and Their Comparison for Custom NER Model.” *Science & Technology Libraries*, 39:3, 324-337, DOI: 10.1080/0194262X.2020.1759479.

Ruokolainen, T. and K. Kettunen. 2020. “Name the Name – Named Entity Recognition in OCRed 19th and Early 20th Century Finnish Newspaper and Journal Collection Data.” *Proceedings of DHN2020*, Riga, March 2020.

Toutanova, K., D. Klein, C. D. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1*. Association for Computational Linguistics. Edmonton, Alberta, Canada, pp. 173–80.

Xue, X. and J. Zhang. 2020. “Evaluation of Seven Part-of-Speech Taggers in Tagging Building Codes: Identifying the Best Performing Tagger and Common Sources of Errors.” *ASCE Library*. Construction Research Congress 2020: Computer Applications (498 - 507).