Jamie Stephens
November 7, 2021

Technology Review

Stanford's Named Entity Recognizer was first introduced in 2005 by the Stanford Natural Language Processing Group, and today is one of the most widely used implementations of all Named Entity Recognizers. These are supervised labeling classifiers that traverse a body of text and identify the names of things. In the technology's most basic form, these entities are limited to well-known people or places, but it can be used to identify entities as diverse as protein or gene names.

Underlying Algorithms

The Stanford NER was built on linear chain Conditional Random Field (CRF) sequence models. CRFs work as a special case of Markov Random Fields - a class of graphical models that graph the space between two variables. CRFs are used to predict any sequence in which multiple variables depend on one another. It considers future outputs while learning new patterns (Dasagrandhi). Labeling is carried out by feature extractors, which can be defined and redefined as needed. This ultimately means that users of Stanford's NER can use the code to build their own sequence models for NER by training custom models on labeled data.

Training Data

This tool was originally trained using the English subset of CoNLL-2003, a dataset composed of words matched with a part-of-speech tag, syntactic chunk tag, and named entity tag. The name originates from the Conference of Computational Natural Language Learning, through which a shared task was assembled to evaluate approaches to named entity recognition. The data was sourced from news articles found in the Reuters corpus and focused on four categories of entity: persons, locations, organizations, and miscellaneous. This last category included entities as diverse as movies, books, holidays, demonyms, and car models. This particular dataset is now considered the gold standard for newswire data in named entity recognition (Augenstein 2017).

Later models of the tool were trained on variants of the MUC 6 and 7 datasets, the acronym referring to the Message Understanding Conference. These datasets also originated from newswire sources; MUC 6 was sourced from approximately 58,000 Wall Street Journal articles between January 1993 and June 1994 (Sundheim 1995), while MUC 7 originates from the New York Times.

Models

Three pre-assembled English language models come with the Stanford NER tool:

- Location, person, organization
- Location, person, organization, misc
- Location, person, organization, money, percent, date, time

The 4-class model was trained on the original CoNLL 2003 dataset, while the 7-class trained on both MUC 6 and MUC 7, and the 3-class model on all of the aforementioned data, including limited amounts of in-house data. All three of the models use distributional similarity measures, which refer to statistical measures that calculate the similarity of any two words given the contexts of word occurrence. Semantic similarity is usually measured using cosine similarity after designating individual words as a vector of values, where each dimension of the vector represents a particular context. These models perform well due to the methods used but occupy considerable space and runtime.

Models exist for the German, Spanish, and Chinese languages, but are not the primary focus of the tool. The Chinese model, notably, requires word segmentation - splitting Chinese characters into text - prior to using the NER. The dependence on English most likely revolves around the fact that most high-profile natural language processing work is still done in an English-speaking context and using English-language datasets, as well as the fact that the tool was developed in an American university.

Applications

The most commonly used application of Stanford's NER is through news article analysis, a predictable use considering the training data used to build the standard models. However, one less conventional use of the Stanford.NLP.NER Java package can be seen in a Personal Identifiable Information (PII) data redaction script that utilizes this same technology to scrub corpuses of data that could be potentially used to identify individuals.

While it was originally released in Java, it also exists in Python's NLTK library as of version 2.0. The latter, notably, means that nltk writes each target sentence into a file each time it's called and then runs the Stanford NER command line tool to parse it, finally parsing the output back to Python. This process unavoidably leads to higher overhead during use (bzdjamboo 2013).

Drawbacks

In one research paper, the Stanford NER was noted to be weaker than other NER systems when faced with generalization and detection of novel entity surface forms - that is, entities that were not included in the training data set (Augenstein et al., 2017). This has been a noted issue with NER systems, whether its limitation came from the specificity of its domain, or other natural limitations arising from its training dataset. Stanford's relative weakness was shown through tests conducted with several corpora where the Stanford NER recorded lower recall rates than counterparts that made use of word embeddings as features; this helps with the unseen word problem.

In a separate work that evaluated Stanford's NER with a NER released by the University of Illinois ("Illinois NER") on a body of Malay-language news articles, the Stanford NER achieved higher precision and F_1 results (39.66% and 36.55%) on a body of 12 documents that held 3,296 words (Mohd Noor 2017). Both the Stanford and Illinois NERs were thought to have performed poorly on two entities, person and miscellaneous, due to differences in morphology between English and Malay.

Conclusion

Stanford's Named Entity Recognizer has its place in the NER space as a highly customizable open-source technology that can be easily adapted to users' needs. While its lack of word embeddings means that its ability to adapt to new datasets is limited - at least relative to its technological counterparts - many natural language processing toolkits today are built as a union of several different systems and pieces of code, creating a Swiss army knife-style of text analysis that negates the need for any one piece to be free of limitations. Future applications of the technology are likely to expand beyond news article syntax and become more widely applicable to specialized domains, including biomedical research and social media applications.

Works Cited

Augenstein, Isabelle, et al. “Generalisation in Named Entity Recognition: A Quantitative Analysis.” *Computer Speech & Language*, vol. 44, 14 Feb. 2017, pp. 61–83., https://doi.org/10.1016/j.csl.2017.01.012.

bzdjamboo. (2013, August 22). *Stanford named entity recognizer (NER) functionality with NLTK*. Stack Overflow. Retrieved November 8, 2021, from https://stackoverflow.com/questions/18371092/stanford-named-entity-recognizer-ner-functionality-with-nltk/19198133#19198133.

Dasagrandhi, C. S. (n.d.). *Understanding named entity recognition pre-trained models*. Blog. Retrieved November 8, 2021, from https://blog.vsoftconsulting.com/blog/understanding-named-entity-recognition-pre-trained-models.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005),* pp. 363-370. http://nlp.stanford.edu/~manning/papers/gibbscrf3.pdf

Mohd Noor, Noorhuzaimi & Sulaiman, Junaida & Mohd Noah, Shahrul Azman. (2016). Malay Name Entity Recognition Using Limited Resources. Advanced Science Letters. 22. 2968-2971. 10.1166/asl.2016.7124.

Poibeau, Thierry; Kosseim, Leila (2001). "Proper Name Extraction from Non-Journalistic Texts" (PDF). Language and Computers. 37 (1): 144–157. doi:10.1163/9789004333901_011. S2CID 12591786. Archived from the original (PDF) on 2019-07-30.

Sundheim, B. M. (1995). Overview of results of the MUC-6 Evaluation. *Proceedings of the 6th Conference on Message Understanding - MUC6 '95*. https://doi.org/10.3115/1072399.1072402