Multilingual Named Entity Recognition with SpaCy and Stanza

A comparison of two NLP tools

Tim Ostrolucký & Shunde Zhang

04.10.2023
Heinrich Heine Universität
Düsseldorf

1 Introduction

The goal of our project is to comparatively investigate the performance, especially accuracy and runtime, of the NLP tools SpaCy and Stanza on Named Entity Recognition (NER). In doing so, we also wanted to gain a better understanding of the two tools and their differences. The results of our project should provide information about the most suitable NLP library for specific situations, and thereby also make clear the advantages and disadvantages of each tool.

To reliably compare the performance of SpaCy and Stanza, we decided to apply the respective NER methods of the two tools to identical data. Furthermore, since we wanted to investigate accuracy and runtime in different languages, we also decided to include Spanish in addition to English. English and Spanish are both languages that we are familiar with, and since they are widely used, we expected to find a lot of (parallel annotated) data for these two languages. In the first step, we compared the accuracy and runtime of the NER methods of the two tools on parallel annotated transcriptions of European Parliament sessions (Europarl corpus). Then, on selected unannotated movie subtitle files in English and Spanish (Open Subtitles), we analyzed the level of concordance in SpaCy and Stanza predictions and what was annotated differently in the two NLP tools. To obtain comparable results for the two libraries, slight modifications had to be made to the tokenizers. Also, since SpaCy uses a different tag set for the annotation of Named Entities than Stanza and the Europarl corpus, we had to transform the results predicted by SpaCy into the other annotation scheme.

Our research has shown that, especially in English, there is a high degree of concordance between the results of SpaCy and Stanza. SpaCy generally provides less accurate predictions than Stanza but has a much shorter runtime. However, some phenomena are not correctly detected by either tool. One has to weigh up carefully whether one prefers the accuracy or the shorter runtime, but in point 4. we explain our results in more detail and try to give hints about which library one should prefer in which situation.

2 Data and Resources

Our main data sources were, on the one hand, the “Evaluation Corpus for Named Entity Recognition using Europarl” and, on the other hand, the website opensubtitles.org.

The Europarl evaluation corpus is a compilation of parallel annotated transcriptions of European Parliament sessions in four languages: German, English, Spanish, and Italian. We focused on the English and Spanish versions. These were manually annotated word by word in CoNLL 2002/2003 format; there are 4 entity types: PER, LOC, ORG, and MISC. The data is contained in the "Data" folder in our GitHub repository, but can also be accessed here. This corpus is especially interesting for our investigations due to its abundance of detailed sentences (it contains almost 800 parallel annotated sentences) and also contains different entity types, such as people, organizations, and places.

On opensubtitles.org, we found the subtitles for our favorite movies in .srt format and had to convert them to .txt format using the website happyscribe so that the data could be processed in our program. Although the subtitles are not annotated, they contain numerous names of people and places, so they are suitable for comparing the predictions of the two NLP tools. In addition, the translations are largely parallel, which ensures comparability in the different languages. The subtitle files are also included in the "Data" folder. We both really enjoy watching movies, and therefore, we wanted to integrate something into the research that is part of our everyday life, bringing more joy and fun into the project. Besides, in contrast to the EU Parliament texts, everyday language is used in these movies.

In terms of software, we used Python version 3.11.4. and the latest versions of the NLP libraries SpaCy (v3.6) and Stanza (1.4.0) available at the time of the project's creation. As SpaCy language models, we took en_core_web_md in English and es_core_news_md in Spanish, which are the models of medium size. We selected SpaCy and Stanza and did not include NLTK in our research because in the case of NLTK, unlike the other two tools we knew from the classroom, we could not find out on which data the language models were trained on. However, a prerequisite for objective measurements was that the language models must not be trained on the data we have chosen. We were able to prove this with SpaCy and Stanza, but not with NLTK.

We also used the re-library for regular expressions, which helped us with some preprocessing steps: inserting spaces before/after punctuation marks in the subtitle files to simplify tokenization, and in the postprocess_labels method, which converted SpaCy's predictions into the 4 entity type system (PER, LOC, ORG, MISC).

3 Method (the code)

utils.py

The Utils Python module defines a set of functions for natural language processing tasks. These functions are designed for processing text data, conducting NER, and evaluating NER performance, specifically tailored to the Europarl and movie subtitle datasets.

Here's a summary of the important aspects of the program:

WhitespaceTokenizer: A custom tokenizer that is provided by the StackOverflow user “Sofia VL”. This tokenizer splits the text on whitespaces, creating tokens and their corresponding spaces. We used it for SpaCy's English and Spanish language models to improve compatibility with the gold data and the Stanza NER results. Since the Europarl corpus consists of a pretokenized text it was necessary to use this custom tokenizer.

For Stanza’s pipeline to split the text on whitespaces, we simply had to add the parameter “tokenize_pretokenized” and set it to true.

load_subtitles(filepath) & load_europarl(filepath): Both load functions, which are created by Tim, do as their name suggests. *load_europarl(filepath)* loads data from a Europarl CONLL02 file, extracting words, labels, and continuous text from the file.

load_subtitles(filepath) loads data from a movie subtitle text file, removing blank lines and line breaks, and processing the text for better tokenization, i.e. add whitespaces before resp. after punctuation marks.

ner(text, model, lang): This is the method in which the actual Named Entity Recognition is being performed. Depending on the model parameter entered, the text is processed with SpaCy or Stanza, returning a list of predicted labels in BIO(ES) format for named entities. We used the BIO(ES) format because the gold data in the Europarl corpus are annotated in this format, and furthermore both tools support this format. Tim did the SpaCy part of the method, while Shunde implemented the Stanza part.

postprocess_labels(pred_labels): Made by Shunde, this method transforms the fine-grained labels predicted by SpaCy into a simplified 4-label format (PER, LOC, ORG, MISC) based on the Europarl data annotation schema (in the docstring is a detailed listing of the changes made to the labels), also used by Stanza.

label_match(label1, label2): The label_match-method, which was implemented by Shunde, checks if two labels are matching, considering various label categories like PERSON (PER), GPE (LOC), ORG, MISC, and O (non-entity).

eval_europarl(word_list, gold_labels, pred_labels, model): The evaluation of the Europarl data takes place in this method – implemented by Tim – by comparing predicted labels with gold labels, calculating the accuracy and identifying differences between them.

eval_subtitles(word_list, SpaCy_labels, stanza_labels): The eval_subtitles-function works similarly: it evaluates the movie subtitle data by measuring concordance between labels predicted by SpaCy and Stanza and identifying differences between them.

el_hoyo.py & back_to_the_future.py

These Python programs, implemented by both Shunde and Tim, perform the Named Entity Recognition in English and Spanish for the movie subtitles. It measures the time it takes for each NER operation with the “perf_counter” method from the “time” module and evaluates the concordance between the labels predicted by SpaCy and Stanza. The results are saved in text files: the ones for the English subtitles are written to a text file (“el_hoyo_en_eval.txt” resp. “back_to_the_future_en_eval.txt”) in the “Evaluation Result” folder, using the latin-1 encoding. The same is done for the Spanish subtitles: the results are saved in “el_hoyo_es_eval.txt” resp. “back_to_the_future_es_eval.txt”.

europarl_en.py & europarl_es.py

These Python programs were created by both of us as well, and they perform Named Entity Recognition (NER) on English or Spanish Europarl data using both the SpaCy and Stanza NER models. By using “perf_counter” from the “time” module it measures the time it takes for each NER operation and it evaluates the accuracy of the predicted labels compared to the gold labels. The results of these evaluations are then saved in separate text files. Fundamentally the structure is the same as that of “el_hoyo.py” and “back_to_the_future.py” with a few changes. The results of the SpaCy evaluation are written in a text file in the “Evaluation Results” folder, in “europarl_en_spacy_eval” resp. “europarl_es_spacy_eval”, while the results of the Stanza evaluation are written in a separate text file (“europarl_en_stanza_eval” resp. “europarl_es_stanza_eval”).

4 Results

After processing data using our program, we came to some interesting conclusions when examining the results. In this section we will first present our measurements, i.e. the quantitative analysis, and then look at some common errors resp. differences between SpaCy and Stanza and investigate their possible causes (qualitative analysis).

Quantitative analysis

To note is that the method “perf.counter” is system-wide. This means that the time we have measured might be (slightly) off from the time that you might have when running the files, depending on the capability of your device. However, these time differences are not all too relevant to the results.

Through multiple tries, we can tell that the results of the evaluations determine that Stanza takes a significantly larger amount of time to process the NER. To highlight one comparison: for the "Back To The Future" English evaluation, Stanza needed roughly *232.392* seconds to finish whilst SpaCy only needed roughly *4.518*. That is about 5000 times longer.

With the given evaluations we can clearly determine that in each tested data source Stanza is by a large margin slower than SpaCy. Furthermore, in the English version of the Europarl corpus, Stanza provides an increase in accuracy of 0.789%; the accuracy for the Spanish version increases by 6,624%. The latter increase might make the wait worthwhile for some, since considering the large data points we are working with: 6,624% of 23279 data points are 1542 wrongly annotated (non) entities. The lack of Spanish accuracy in SpaCy can also be seen in the evaluation of the subtitles. In these, SpaCy made a large number of sometimes strange errors, for which we could not always find out the reason.

In the following charts, all comparisons are listed with the time required in each case and the time differences in percent, as well as the accuracy resp. concordance of the two NLP tools.

Data Source	*SpaCy runtime (in sec)*	*Stanza runtime (in sec)*	*Difference (in %)*
back_to_the_future_en	4.518	232.392	~ 5043%
back_to_the_future_es	4.152	220.313	~ 5206%
el_hoyo_en	3.064	97.466	~ 3078 %
el_hoyo_es	1.977	99.637	~ 4937.7 %

[Chart 1] Runtime comparison of SpaCy and Stanza on the subtitle files

Data source	*Total data points*	*Concordant predictions*	*Concordance (in %)*
el_hoyo_en	7427	7288	98.128 %
el_hoyo_es	7032	5621	79.935 %
back_to_the_future_en	13085	12798	97.807 %
back_to_the_future_es	11464	9346	81.525 %

[Chart 2] Concordance between Spacy and Stanza on the subtitle files

Data Source	*Tool*	*Total data points*	*Correct predictions*	*Accuracy (in %)*	*Runtime (in sec)*	*Runtime difference (in %)*
europarl_en	SpaCy	22320	21407	95.909 %	~7.473	5443.87%
	Stanza		21583	96.698 %	~413.985	
europarl_es	SpaCy	23279	20662	88.758 %	~7.12	7686.52%
	Stanza		22204	95.382 %	~554.904	

[Chart 3] Accuracy and runtime of SpaCy and Stanza on the Europarl corpus

Qualitative analysis

To further expand the evaluation of the differences of SpaCy and Stanza we also have manually analyzed around one hundred differences in each evaluation and condensed them to the problems and special cases that we found most worth mentioning.

Europarl EN

The main deviations that the NLP tools show from the gold data are with entities that are context-sensitive to the gold data. A few examples would be "95/35/EC" or the "directive of the transport of dangerous goods by road", both names of legal texts that were not recognized correctly, or the "Conference of Presidency" and "Member States" both being labeled as an organization (ORG) in the Europarl corpus while Stanza and SpaCy labeled them as miscellaneous (MISC). One reason for this could be that in the (training data of the) language models we used, these entities in their context were usually not referred to as ORG.

Additionally, both SpaCy and Stanza have the habit of annotating the salutations "Mr" and "Mrs" as PER, if coming before a name; SpaCy has trouble labeling the names of "Prodi" and "De Palacio" as PER, but instead labeling "Erika", the hurricane, and "margarine" as PER.

Another problem seems to be that some words, like "[the] Group" and "Structural Funds (program)", which are written in capital letters are being labeled by Stanza and SpaCy as an organization. It appears to be that the NLP, whatever that it can not properly label and is in capital letters, is being labeled as an organization instead.

Overall it seems that SpaCy makes a (few more) mistakes than Stanza in the NER on the English Europarl corpus.

Europarl ES

The Spanish NER of Stanza on the Europarl shares the same problems and differences as the English NER has with labeling. A unique case in the labeling is that of Russia being seen as an organization rather than a LOC, as well as declaring all other country adjectives, like “neerlandés” (= Dutch) or “español” (= Spanish), non-entities (O) instead of MISC. Also, it won’t recognize the Italian name “Prodi”.

Whilst SpaCy shares the same errors and differences with the Spanish Stanza, seemingly inexplicable mistakes occur as well: whole sentences (like “Una de las personas” [= one of the persons] and “Como todos han podido comprobar” [= as everyone has been able to see]) and words (like “Invito” [= I invite]) are being labeled as MISC by SpaCy. In none of these sentences can there be anything found that we would consider to be an entity. Moreover, SpaCy makes mistakes in labeling persons by recognizing “Señorías” (= Ladies and Gentlemen) and “Sabrá” (= he will know) as PER.

Lastly, it is also noteworthy that neither of the NLP tools seems to be able to recognize a Named Entity that is a significant event in world history, such as the "millennium bug"/"efecto del año 2000", in either English or Spanish.

Back To The Future

Overall in both English and Spanish versions, SpaCy struggles with the labeling of PER to entities. Most of the time these words are clearly non-entities, such as “don” (from “don’t”), “gonna”, “Re-elected”, and “Statler Toyota”. Whilst well-known names like “Einstein” are being overlooked and labeled “O”. Stanza, though not nearly as frequent as SpaCy, does also share this issue and labels words such as “huh” and “hello” as PER. To note here is that after these words follow an entity labeled as PER or an imperative sentence.

A special entity is “Doc”, the nickname of Doctor Emmett Brown, one of the main characters. Both tools seem to struggle in labeling him, since he’s sometimes labeled as PER and sometimes disregarded completely. But yet again, SpaCy tends to be the one that struggles more often, although not only with the entity “Doc”, but also with specific names of the fictional characters, events and locations, e.g. “Twin Pine Mall”, the “Fish under the Sea” ball, “Jailbird” or “Pinheads”. Stanza on the other hand has no problems in labeling them correctly, except with the “DeLorean” (real-life car model which the time machine of Doctor Emmett Brown is based on); this Stanza is unable to do but SpaCy identifies it correctly.

Just as in the NER with the Spanish Europarl, SpaCy is performing odd labeling of whole sentences and words that are not entities. An example would be: “Espera un minuto” (= wait a minute) and “Un grupo terroista libio” (= a Libyan terrorist group) which are both labeled as MISC. This occurs very frequently. In contrast, this is an uncommon event for Stanza.

To summarize: Stanza shows a higher degree of accuracy and seems to have a vast knowledge of fictional entities, although struggling with identifying PER Named Entities and infrequently labeling whole sentences wrong. SpaCy on the other hand is more prone to mistakes, in both mislabeling PER and more than frequently labeling whole sentences wrong.

El Hoyo

Yet again, SpaCy is mislabeling whole sentences and words as MISC.

The NER on the Spanish El Hoyo file is the one with the highest amount of differences. This is mostly due to SpaCy’s mistakes with the mislabeling of whole sentences. In the approximately one hundred differences we manually analyzed, almost every third instance was SpaCy mislabeling a whole sentence. The other mistakes are SpaCy labeling “qué” (= what) as MISC, “obvio” (= obvious) as MISC, PER and LOC, “Dios” (= God) as MISC, PER, and O, and also not being able to recognize the names of some of the characters, “Goreng” and “Trimagashi”, labeling them as LOC or O. To note here is that these names can be harder to decipher, given that these are uncommon names in Indonesian. One means “fried” whilst the other means “thank you”.

Stanza does not have this kind of problem. The only error is, this time with a wider variety of entities, wrongly labeling them as PER. Some of them are (English examples):

“Vertical self management center” | *“Okay”* | *“Messiah”* | *“Bastards”* | *“Isn“* | *“Hotter”* | *“Aren”* |
“Hey” | *“F*ck”* | *““Together”* |

Despite SpaCy having a smaller variety of words that are wrongly labeled as PER, these mistakes come about more regularly, e.g.:

“Don” | *“isn”* | *“ma”* | *“Drop”* | *“Panna”* | *“sin-ridden”* | *“Wake”* | *“loin”* | *“Samurai-Max”*

A special case that appeared when labeling the Spanish El Hoyo subtitles is how “facial” is being labeled. It appears to be that SpaCy split the word into “fa”, “cial”, and “.”, then labeled them all as PER. That SpaCy splits words is a habit that appears only twice and this is only if “facial” ends with a period, else it is not split and correctly labeled.

Finally, in all instances, SpaCy is acting atheistic, as it sees the capitalized God and Christ as a non-entity, while Stanza sees them as PER. Which might be also sacrilegious depending on how one sees it.

5 Challenges and Open Issues

While making the project, we faced some challenges, most of which we were able to solve ourselves. For example, it was difficult to ensure that both SpaCy and Stanza split the text data in the same way (as in the gold data) in order to compare and thus evaluate the predictions of the NLP tools with each other or with the manual annotations. We achieved this for Stanza by setting the “tokenize_pretokenized” parameter to True and for SpaCy by using a custom tokenizer.

In addition, we had to “post-process” SpaCy's predictions because they have a different format than the gold labels and the Stanza predictions. To do this, we manually examined the gold data to see which labels in the SpaCy tagset the manual annotations corresponded to, and then wrote a method to perform the conversion from SpaCy labels to the 4-label format.

Other problems, however, were more difficult to solve or remained as open issues. For example, it was more difficult than anticipated to find parallel annotated corpora for Named Entity Recognition in a format that could be easily processed using a Python program. The "Evaluation Corpus for Named Entity Recognition using Europarl" was the most suitable data source for this purpose that we could find, even though the language in it may not be entirely representative of everyday language.

In future research, the NLTK NLP tool could be included in the comparison after ruling out the possibility that it was trained on the data we used. However, we could not rule this out based on the data available on the internet.

6 Summary and Conclusion

It is clear to see that manual observations on annotations are highly advised, given that (both) NLP tools can create inexplicable predictions. These errors and differences that we have found and that are mentioned are intentionally explained in a concise manner so that one can get an overview of the most common sources of errors. If we had more time, we would have further analyzed the cause of their occurrence, but this will be left for another time or later for others interested.

To summarize our findings:

SpaCy has proven to be significantly faster than Stanza. In some cases up to 5000% faster. If English data is used for the NER, SpaCy is to be preferred because the much shorter runtime outweighs the only slightly improved accuracy of Stanza (less than 1% higher).

However, if you use Spanish data or subtitles for Named Entity Recognition, you should use the Stanzas model despite the longer runtime. In our research we found that using SpaCy with Spanish data results in a large number of errors compared to Stanza and also many seemingly inexplicable labels. In addition, Stanza offers extensive knowledge and/or recognition of fictional names, which applies to both English and Spanish data.

7 Addendum: Running our code with a GPU

After running our code in Google Colab with a GPU, we came up with the following results:

Data Source	*SpaCy runtime (in sec)*	*Stanza runtime (in sec)*	*Difference (in %)*
back_to_the_future_en	1.74	14.468	831.494
back_to_the_future_es	1.35	12.809	948.815
el_hoyo_en	2.078	7.948	382.483
el_hoyo_es	0.885	7.563	854.576
europarl_en	4.631	30.32	654.718
europarl_es	4.37	31.502	720.87

[Chart 4] Runtime comparison of SpaCy and Stanza after running the code with a GPU

Using a GPU speeds up the Named Entity Recognition process, especially in Stanza, where there is a significant performance improvement; in SpaCy, the difference in runtime is rather small in comparison. Nevertheless, SpaCy is still 4 - 10 times faster, with the same accuracy resp. concordance of the two NLP tools, which means that our findings remain valid.

The runtime difference can be explained by the different architectures of the NLP libraries. While SpaCy uses models pre-trained with CNNs, Stanza is based on bidirectional LSTMs. Due to their complexity and the number of parameters, LSTMs require a higher computational power than other types of neural networks, and hence the longer runtime from the Stanza NER.

8 Sources

Agerri, R., Chung, Y., Aldabe, I., Aranberri, N., Labaka, G. & Rigau, G. (2018). Building Named Entity Recognition Taggers via Parallel Corpora. In *Proceedings of the 11th Language Resources and Evaluation Conference (LREC 2018)*.

Honnibal, M., Montani, I., Van Landeghem, S. & Boyd, A. (2020). *2020. SpaCy: Industrial-strength Natural Language Processing in Python*.

ixa-ehu. (n.d.). *ner-evaluation-corpus-europarl: Manually annotated test set from Europarl for Named Entity Recognition*.

Koehn, P. (2005). *Europarl: A Parallel Corpus for Statistical Machine Translation*.

Qi, P.; Zhang, Yuhao; Zhang, Yuhui; Bolton, J. & Manning, C. D. (2020). Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Association for Computational Linguistics (ACL) System Demonstrations*.

SpaCy · industrial-strength Natural Language Processing in Python. (n.d.). SpaCy.Io. Retrieved September 12, 2023, from https://SpaCy.io/

Stanza – A Python NLP Package for Many Human Languages. (n.d.). Stanza. Retrieved September 12, 2023, from https://stanfordnlp.github.io/stanza/

Subtitle converter. (n.d.). Happy Scribe. Retrieved September 12, 2023, from https://www.happyscribe.com/de/untertitel-tools/untertitel-converter

Subtitles - download movie and TV Series subtitles. (n.d.). Opensubtitles.org. Retrieved September 12, 2023, from https://www.opensubtitles.org/

Trained Models & Pipelines · SPACY Models Documentation. (n.d.). Trained Models & Pipelines. https://spacy.io/models

What are the disadvantages of LSTM compared to CNN/RNN? (n. d.). Quora. https://www.quora.com/What-are-the-disadvantages-of-LSTM-compared-to-CNN-RNN