Heinrich-Heine-Universität Düsseldorf
Institut für Sprache und Information
SoSe 2022
Aufbauseminar: Advanced NLP with Python
Herr Dr. Kilian Evang

Named Entity Recognition in English, German, and Dutch with spaCy and Stanza

Julia Hillebrand
julia.hillebrand@hhu.de
B.A. Computerlinguistik, Fachsemester: 6
2844565
06.10.2022

Table of Contents

1. Introduction

The goal of Named Entity Recognition (NER) models is to accurately identify and classify named entities into pre-defined categories in a text. An example of named entity recognition is the following, using the NLP toolkit spaCy[1] for identification and tagging (Honnibal et al., 2020):

> *Microsoft* [ORG] was founded by *Bill Gates* [PERSON] and *Paul Allen* [PERSON] on *April 4, 1975* [DATE] in *Albuquerque* [GPE], *New Mexico* [GPE], *U.S* [GPE].

Many Natural Language Processing (NLP) tools for NER exist today, for a multitude of languages, and their performance varies by language and tool. In this project, I evaluated the performance of SpaCy and Stanza[2] (Qi et al., 2020) for English, German, and Dutch based on efficiency (i.e., time), accuracy (precision, recall, F1-score), and ease of implementation on highly similar texts to determine which of the two is best to use for NER based on these criteria.

2. Methods

For this project, I used Tatoeba[3] (released under a CC-BY 2.0 FR license) as a source for the to-be annotated sentences and spaCy's and Stanza's pipelines for named entity tagging and evaluation.

2.1 Data

All data used in this project is from Tatoeba.

I decided against using literature as a source since I suspected the occurring named entities might not vary enough and I wanted to determine the performance across many sentences with a variety of named entities, not repeat the same named entities over and over.

[1] https://spacy.io/
[2] https://stanfordnlp.github.io/stanza/
[3] https://tatoeba.org/en/

2.2.1 Tatoeba

Tatoeba is a collection of sentences and their translations.

There are two main reasons I decided to use Tatoeba. First of all, it was not possible for me to get access to a parallel corpus that was open source and happened to have data for English, German, and Dutch simultaneously. I also wanted sentences that are similar in structure for comparison purposes. If a translation exists in Tatoeba, the sentences tend to be similar in structure and word choice, making it easier to compare where spaCy and Stanza make mistakes and also easier to see if the pipelines produce similar results in each language.

Tatoeba's downside is that the number of sentences changes by language since some sentences are not yet translated or only into two languages, so it is not a true parallel corpus.

Table 1: Corpora statistics.

	English	German	Dutch
Total sentences	1624792	591234	163112
Used sentences	First 163112	First 163112	163112

Due to technical limitations, I tagged and evaluated 163112 sentences per language which corresponds to the number of Dutch sentences available as of August 15, 2022.

2.2 Tools

2.2.1 SpaCy

"spaCy is a free, open-source library for advanced Natural Language Processing (NLP) in Python. It's designed specifically for production use and helps you build applications that process and "understand" large volumes of text. It can be used to build information extraction or natural language understanding systems." (spaCy, n.d.-a, Facts & Figures).

I chose spaCy v3 for comparison because it is openly accessible, beginner-friendly, and we learned about it in class.

SpaCy provides pre-trained pipelines for all three languages, including a multi-language pipeline specifically for NER. I used the multi-language pipeline for tagging and evaluating, the md (medium) pipeline (XX2 in this project, XX being the language code) for tagging, and the lg (large) pipeline (XX3 in this project) for evaluation because it provides slightly better accuracy. It is slower than the md pipeline due to its size. See Table 2 below for the exact accuracies (spaCy, n.d.-b, n.d.-c, n.d.-d, n.d.-e).

Table 2: Accuracy per pipeline in spaCy.

Note. The data is taken from spaCy (n.d.-b, n.d.-c, n.d.-d, n.d.-e).

	Dutch	English	German	Multi-language
Precision	Md: 0.76 Lg: 0.78	Md: 0.86 Lg: 0.86	Md: 0.85 Lg: 0.86	0.83
Recall	Md: 0.74 Lg: 0.76	Md: 0.85 Lg: 0.85	Md: 0.85 Lg: 0.85	0.83
F-score	Md: 0.75 Lg: 0.77	Md: 0.85 Lg: 0.86	Md: 0.85 Lg: 0.86	0.83

The pipelines process data as seen in Figure 1 below. In spaCy v3, the NER component operates independently from the other components with its own tok2vec component. Therefore, the rest of the pipeline is unnecessary for NER (spaCy, n.d.-f).

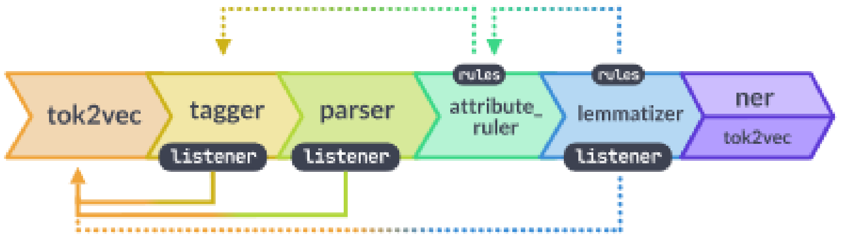

Figure 1: How a spaCy pipeline work after given a text.

Note. The figure is taken from spaCy, n.d., (https://spacy.io/models#design).

For English and Dutch, spaCy uses eighteen named entity types for classification: CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP, ORDINAL, ORG, PERCENT, PERSON, PRODUCT, QUANTITY, TIME, and WORK_OF_ART.

German and the multi-language pipeline provide four types: LOC, PER, ORG, and MISC.

For a detailed explanation of each type, take a look at Appendix A.

2.2.2 Stanza

Stanza is an open-source Python natural language processing toolkit. It provides a fully neural pipeline for text analysis which includes tokenization, multi-word token expansion, lemmatization, part-of-speech and morphological feature tagging, dependency parsing, and named entity recognition (Qi et al., 2020).

I chose Stanza for the same reasons as spaCy: it is open source, beginner-friendly, and it was discussed in class.

Stanza provides one pipeline and two NER models per language. For English, I used the model trained on OntoNotes as it comes with the same eighteen categories as spaCy. For German, I used GermEval2014 and for Dutch CoNLL02, the automatically selected models. Unfortunately, both Dutch models only provide four named entity types: LOC, PER, ORG, and MISC (Stanza, n.d.-a). The German models do the same but, in this case, it does not interfere with the later analysis since spaCy and Stanza share the same types.

The F1-score for each NER model is: Dutch: 89.2; English: 88.8; German: 85.2 (Stanza, n.d.-a).

A detailed explanation of each type can, again, be found in Appendix A.

Stanza processes data as seen in Figure 2. For NER only the TOKENIZE and NER processors are necessary.

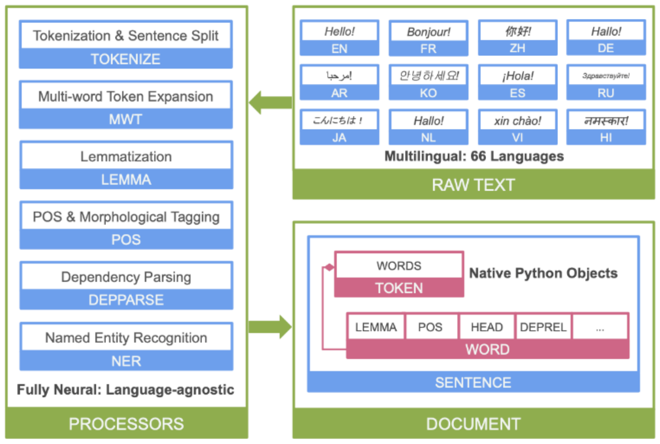

Figure 2: How a Stanza pipeline works.

Note. This figure is taken from Stanza – A Python NLP Package for Many Human Languages, n.d., (https://stanfordnlp.github.io/stanza/index.html).

2.3 Implementation

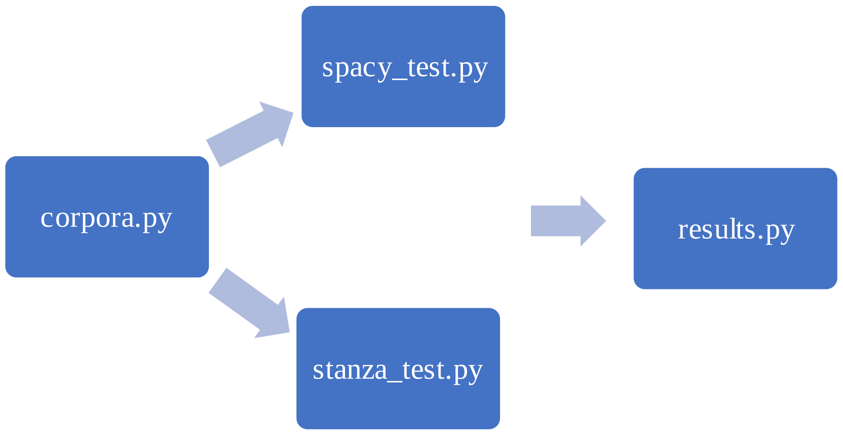

Figure 3: Module structure.

Figure 3 shows the module structure. The project uses four modules. A corpora extraction module *corpora.py*, two modules *spacy_test.py* and *stanza_test.py* where the corpora are annotated with named entities and then evaluated, and the final module *results.py* which prints the results.

2.3.1 corpora.py

In this module, the three TSV corpus files are opened, and the indexing and language tag are removed from the entries. The module returns the three corpora as three ready-to-use lists.

2.3.2 spacy_test.py

The purpose of spacy_test.py is to find, tag, and later evaluate all named entities in the corpora with spaCy.

It consists of a part that downloads the pipelines as well as two functions.

The first function *named_entity_recognition_sp* locates named entities in all sentences and assigns the entity types, using the lg or multi-language pipeline.

The second function *evaluate_ner_sp* evaluates the accuracy (precision, recall, F1-score) of the result from *named_entity_recognition_sp* with spaCy's lg pipeline.

2.3.3 stanza_test.py

Stanza_test.py has the same structure as spacy_test.py.

Due to large processing times, I limited the Stanza pipelines to tokenization and NER.

The *named_entity_recognition_st* function in this module assigns the entity types using Stanza's pipelines. Thereby each corpus is only tagged once.

The *evaluate_ner_st* function evaluates with spaCy's lg pipeline, like spacy_test.py, but also with the multi-language pipeline. I chose to add the multi-language pipeline because of the entity type difference between spaCy and Stanza for Dutch discussed in section 2.2.2.

2.3.4 results.py

In the results.py module, the functions defined in spacy_test.py and stanza_test.py are called and the functions' results printed. It also has an *export_to_csv* function which uses pandas to make the results more pleasant to read.

3. Results

The accuracy of the results varies by language, depending on the pipeline used for tagging and evaluation as well as the toolkit.

SpaCy and Stanza are quick to install with a pip-command. Their pipelines are installed the same way. In Stanza they can even be downloaded from within modules. Both toolkits work directly on the text (though in Stanza any list first has to be converted into a Stanza.Document). Their documentations explain sufficiently how to use both toolkits and their implementation is intuitive. Both toolkits come with a command line evaluation command, but it is not compatible with the data used in this project. Thus, it was not used.

Table 3: SpaCy's accuracy and time, total (rounded).

Tagger: Evaluation:	nlp_de2 nlp_de3	nlp_multi nlp_de3	nlp_en2 nlp_en3	nlp_multi nlp_en3	nlp_nl2 nlp_nl3	nlp_multi nlp_nl3
Precision	0.6558	0.5485	**0.9261**	0.0274	0.8607	*0.0051*
Recall	0.5094	0.3471	**0.8877**	0.0237	0.7921	*0.0047*
F1-score	0.5734	0.4251	**0.9065**	0.0254	0.823	*0.0049*
CPU time total (in seconds)	*889.4063*	541.4375	740.2188	**357.875**	807.875	431.6875

Table 4: Stanza's accuracy and time, total (rounded).

Tagger: Evaluation:	nlp_de nlp_de3	nlp_de nlp_multi	nlp_en nlp_en3	nlp_en nlp_multi	nlp_nl nlp_nl3	nlp_nl nlp_multi
Precision	0.5275	0.5834	**0.8896**	0.0312	*0.0083*	0.6935
Recall	0.4555	0.5637	**0.8289**	0.0169	*0.0082*	0.6531
F1-score	0.4889	0.5734	**0.8582**	0.0219	*0.0082*	0.6727
CPU time total (in seconds)	24121.5625	*24507.328125*	23592.46875	23334.15625	**18424.796875**	18476.90625

During the evaluation, a handful of sentences in spaCy and depending on the language around 100-200 sentences in Stanza were ignored due to misalignment (the start and end characters did not match where the entity was supposed to end; it seems that this is an error caused during training by punctuation which cannot be fixed manually).

English scored the highest F1-score for both toolkits (spaCy: 90.65%, Stanza: 85.82%) as seen in Table 2 and Table 3. At the same time, English received the second lowest F1-score both in Stanza (2.19%) and spaCy (2.54%). Only Dutch performed worse at 0.49% (spaCy) and 0.82% (Stanza) accuracy for F1. While Dutch received the overall worst F1-score, it also achieved the second-best F1-score in both toolkits (spaCy: 82.3%; Stanza: 67.27%). In German, Stanza performed a little better overall, but the F1-score stayed in the 40-60% accuracy zone in Stanza and spaCy, placing German in the middle of the scale.

Recall produced worse results than precision in all cases. This means that there were more false negatives (recall) than false positives (precision). In other words, more named entities were not recognized as named entities than non-named entities were recognized as named entities. Generally, the recall score was about 10% lower than the precision score (except

German in spaCy). Since the F1-score is the harmony between recall and precision the same accuracy order as for the F1-score applies.

Accuracy in all categories decreased (though for the F1-score this is by-effect) significantly when the entity types did not match. I.e., when any combination occurred where an eighteen-class pipeline was paired with a four-class pipeline. This is likely because the evaluating pipeline cannot recognize entity types outside the ones it was trained for. The category automatically scored zero. Though even for the types it should recognize, the score only surpassed 50% once, in precision (Stanza: nlp_nl, nlp_nl3, LOC: p: 0.8006872852233677), and never for recall.

Though Stanza did perform better by around 15% in German when the multi-language pipeline was used, spaCy overall had the higher F1-score if you ignore the multi-language pipelines for English and Dutch which essentially had an accuracy of zero in all categories. As the F1-scores already suggest, precision and recall are also better in spaCy (excluding the German result mentioned before).

SpaCy ran faster than Stanza in all instances. The longest time elapsed for spaCy was 889.4063 seconds/ ca. 15 minutes in CPU time compared to Stanza's fastest model which took 18424.796875 seconds/ ca. 5 hours in CPU time to complete. Windows 10 with an AMD Ryzen 5 3600 6-Core Processor was used.

For the statistics per entity type, please see Appendix B.

In general, none of the scores per entity type stood out as particularly high or low. If the pipelines' categories did not match, the total accuracy decreased as mentioned above and therefore, the score per entity type, too. In the four-class pipelines, MISC and ORG tended to have low scores for all pipelines (ignoring mismatched pipelines). In the eighteen-class pipelines, FAC, ORG, PRODUCT, LAW, and WORK_OF_ART overall performed worse than the other types in relation to the total accuracy of the trained data evaluated. In both instances, this might be because the pre-trained models used in the evaluation were not sufficiently trained in these categories or because language-specific entities were mentioned in the text and not accurately translated. The rest of the types scaled to the total accuracy of the pipeline, deviating by around 5% above and 15% below the scores of total precision, recall, and F1-score. An F1-score of +90% reached GPE and ORDINAL across all combinations (excluding mismatched pipelines). In the four-class pipelines, there was no clear most accurate type, but given the earlier-mentioned results, either LOC or PER took the top spot with noticeably better scores than MISC and ORG.

In German with spaCy nlp_de2 mistakes occurred frequently in shorter sentences (below ten words) and at the start of sentences. Often the sentence was falsely assigned MISC or ORG. This happened mainly when the sentence started (i) with a determiner or pronoun—in that case, the following verb was tagged too—or (ii) the sentence was an imperative or exclamation—in this case, the entire sentence tended to be tagged as one named entity. There was also a noticeable issue where the model identified parts of a sentence as a named entity—even when there was no named entity—and then split the sentence part into two separate named entities ("*Du hast nie* [MISC] *Zeit* [MISC] für die wichtigen Dinge!"). This seemed to be arbitrary. Sometimes spaCy correctly located a named entity but assigned the wrong label e.g., "Japaner" as LOC instead of MISC. In other cases, the model identified nouns as named entities that are only named entities in an extended sense like "Heute Abend gehen wir in die *Kirche* [ORG]." or job titles like "Kommissar [PER]" and not named entities. The model often mistakenly tagged constructions that could be used with people as named entities, which also resulted in common nouns and their articles to be falsely tagged, for example "*Der Wind* [MISC] hat sich beruhigt." The model also struggled to identify names. In particular, it identified adjectives and imperatives at the start of the sentence as PER. Names that are uncommon in German might be missed entirely like "Muiriel".

In German with nlp_multi the same issues occurred. This model also tagged interrogatives as PER and pronouns as MISC.

Much like in German, the nlp_en2 struggled with names and skipped them or assigned the wrong category. It also did not produce reliable results for WORK_OF_ART, e.g., tagging "Mathematics" as a WORK_OF_ART. However, otherwise, this model made few mistakes in tagging once it identified a named entity and unlike the German models, did not have issues identifying word classes correctly.

Nlp_multi in English suffered from the same issues as the German models, though it did not misidentify word classes as frequently as in German. Still, the same issue of tagging entire exclamations as a named entity or the first word for seemingly no reason arose in noticeable numbers. It skipped named entities entirely, in particular in the MISC category, even when other named entities in the sentence were correctly identified. Alternatively, the model assigned the wrong type when MISC was available. The model also was not able to differentiate homonyms, e.g., the interrogative who in “Who is it?” was identified as the organization WHO and thus tagged as ORG.

With nlp_nl2, names were again an issue. Either they were skipped or assigned the wrong category. The WORK_OF_ART category stood out with many mistakes where common nouns or the entire sentence were tagged as WORK_OF_ART. Like in the previous models, there were also a few instances of the first word of the sentence being tagged falsely (commonly as PERSON). However, overall, the results using nlp_nl2 were correct.

While the nlp_multi model on Dutch had the same issues as the previous instances with this model, it performed noticeably better here. Particularly, the LOC category showed few mistakes. Though the issue of tagging entire sentences and interrogatives as named entities occurred again. Names also were ignored or were assigned the wrong category. Generally, this instantiation of nlp_multi produced the most accurate results, despite what the earlier evaluation suggested.

In Stanza, the German and Dutch modules had near-perfect accuracy compared to their spaCy counterparts, but there were still a few times when the modules did not recognize dates and cardinal numbers at all when they should have been MISC. The English module could successfully overcome these issues, making the results essentially perfect. Errors occurred when sentences were ambiguous, and a named entity could have occurred in the position of the common noun.

4. Discussion

Overall, the results in accuracy while using Python for evaluation were low. However, an evaluation by hand to showcase where spaCy and Stanza made mistakes, showed that Stanza produced significantly better results than the Python evaluation led to believe. This makes the evaluation results hard to believe and raises issues with the evaluation method used. If one were to purely believe the Python evaluation, then spaCy would be the best option since it was faster and gained higher scores.

At the moment, all results are evaluated with pipelines that do not have perfect accuracy themselves or in some cases do not even work well together with the training pipeline due to entity type differences. Only spaCy being able to evaluate the training results is also an issue. Likely, it would have been better to annotate text with a set number of entity types by hand and then let Stanza and spaCy annotate that text before using the hand-annotated text as an NLP model for evaluation (though this was not possible for this project due to time constraints and lack of text availability). Under the current circumstances, you cannot draw any exact conclusions about spaCy's and Stanza's accuracy since the results are largely conjecture.

Because of the evaluation method used neither spaCy nor Stanza ever reached the accuracy given on their websites, making it impossible to compare their values and see how well spaCy and Stanza performed in terms of accuracy (spaCy, n.d.-b, n.d.-c, n.d.-d, n.d.-e; Stanza, n.d.-a).

Ultimately, whether you want to use spaCy or Stanza depends on your needs. Neither toolkit is difficult to implement, especially if one simply wants to tag a sentence or text and does not want to evaluate the performance of the two (since you have to implement an evaluation method yourself, depending on the training model and evaluation model you want to use). Stanza is more accurate, as the by-hand evaluation revealed, but also takes a significantly longer time to run. SpaCy, on the other hand, runs faster but also has lower accuracy, in particular in German. Unlike Stanza, spaCy also provides more entity types for Dutch which could be a reason to favor spaCy.

While the research questions could be answered, the answers are not satisfactory and demand further research. In the future, I would like to use this implementation with the suggested solution above to see whether it is possible to improve the accuracy and if the accuracy comes closer to the projected pipeline accuracy by spaCy and Stanza. At the moment, the results.py file is also a little cumbersome to use since it requires a lot of manual input. This is another area I want to improve. I also would like to expand the corpora.py file to make it possible to include other corpora of other formats.

References

Honnibal, M., Montani, I., Van Landeghem, S. & Boyd, A. (2020). *spaCy: industrial-strength natural language processing in Python*. https://doi.org/10.5281/zenodo.1212303.

Qi, P., Zhang, Y., Zhang, Y., 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*. https://arxiv.org/abs/2003.07082.

spaCy · Industrial-strength Natural Language Processing in Python. (n.d.-a). *Facts & Figures*. Retrieved September 24, 2022, from https://spacy.io/usage/facts-figures.

spaCy · Industrial-strength Natural Language Processing in Python. (n.d.-b). *English*. Retrieved September 24, 2022, from https://spacy.io/models/en.

spaCy · Industrial-strength Natural Language Processing in Python. (n.d.-c). *German*. Retrieved September 24, 2022, from https://spacy.io/models/de.

spaCy · Industrial-strength Natural Language Processing in Python. (n.d.-d). *Dutch*. Retrieved September 24, 2022, from https://spacy.io/models/nl.

spaCy · Industrial-strength Natural Language Processing in Python. (n.d.-e). *Multi-language*. Retrieved October 4, 2022, from https://spacy.io/models/xx.

spaCy · Industrial-strength Natural Language Processing in Python. (n.d.-f). *Trained Models & Pipelines*. Retrieved September 26, 2022, from https://spacy.io/models#design.

Stanza: A Python NLP Library for Many Human Languages. (n.d.). Retrieved September

23, 2022, from https://stanfordnlp.github.io/stanza/.

Stanza: A Python NLP Library for Many Human Languages. (n.d.-a). *NER Models*. Retrieved September 24, 2022, from https://stanfordnlp.github.io/stanza/ner_models.html.

Tatoeba. (n.d.). *Tatoeba: collection of sentences and translations*. Retrieved September 29, 2022, from https://tatoeba.org/en/.

Appendix A

Named entity labels by language:

- German: LOC, MISC, ORG, PER
- English: CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP, ORDINAL, ORG, PERCENT, PERSON, PRODUCT, QUANTITY, TIME, WORK_OF_ART
- Dutch: CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP, ORDINAL, ORG, PERCENT, PERSON, PRODUCT, QUANTITY, TIME, WORK_OF_ART
- Multi-language: LOC, MISC, ORG, PER

Explanation of each entity type:

- LOC: Non-GPE locations, mountain ranges, bodies of water
- MISC: Miscellaneous entities, e.g., events, nationalities, products or works of art
- ORG: Companies, agencies, institutions, etc.
- PER: Named person or family.
- CARDINAL: Numerals that do not fall under another type
- DATE: Absolute or relative dates or periods
- EVENT: Named hurricanes, battles, wars, sports events, etc.
- FAC: Buildings, airports, highways, bridges, etc.
- GPE: Countries, cities, states
- LANGUAGE: Any named language
- LAW: Named documents made into laws.
- MONEY: Monetary values, including unit
- NORP: Nationalities or religious or political groups
- ORDINAL: "first", "second", etc.
- PERCENT: Percentage, including "%"
- PERSON: People, including fictional
- PRODUCT: Objects, vehicles, foods, etc. (not services)
- QUANTITY: Measurements, as of weight or distance

- TIME: Times smaller than a day
- WORK_OF_ART: Titles of books, songs, etc.

Appendix B

Table 5: A table of the results obtained during evaluation with Python in detail.

spaCy	Stanza
Nlp_de2, evaluation: nlp_de3 • **'ents_p': 0.6557663584072203, 'ents_r': 0.5094244133657562, 'ents_f': 0.5734054895204485,** • **'ents_per_type':** • **'MISC': {'p': 0.5710301085129146, 'r': 0.4597188470884986, 'f': 0.5093641894309914},** • **'ORG': {'p': 0.5495742092457421, 'r': 0.5101637492941841, 'f': 0.5291361639824304},** • **'PER': {'p': 0.6978459377195037, 'r': 0.4942581153351851, 'f': 0.5786676374226429},** • **'LOC': {'p': 0.7685173886516168, 'r': 0.6043855861043136, 'f': 0.6766404340468963}** • 889.40625s	Nlp_de, evaluation: nlp_de3 • **'ents_p': 0.5274787923013774, 'ents_r': 0.4555408944994781, 'ents_f': 0.4888776167982331,** • **'ents_per_type':** • **'ORG': {'p': 0.2810975609756098, 'r': 0.2962724935732648, 'f': 0.28848560700876097},** • **'PER': {'p': 0.6954373728567277, 'r': 0.36911923492210397, 'f': 0.48226521563885527},** • **'LOC': {'p': 0.7432432432432432, 'r': 0.5912953008496619, 'f': 0.6586190246257846},** • **'MISC': {'p': 0.19531462816671208, 'r': 0.4789579158316633, 'f': 0.2774767801857585}** • 24121.5625s
Nlp_multi, evaluation: nlp_de3 • **'ents_p': 0.5484837509466267, 'ents_r': 0.34712483199099786, 'ents_f': 0.42516861388855215,** • **'ents_per_type':** • **'LOC': {'p': 0.7469879518072289, 'r': 0.3608992372541148, 'f': 0.48666937339288135},** • **'ORG': {'p': 0.41804043545878694, 'r': 0.2181110029211295, 'f': 0.2866588461128292},** • **'PER': {'p': 0.6724708340588542, 'r': 0.31869068903864667, 'f': 0.43244252015527024},**	Nlp_de, evaluation: nlp_multi • **'ents_p': 0.5833685099197974, 'ents_r': 0.5636981645139361, 'ents_f': 0.573364679850643,** • **'ents_per_type':** • **'ORG': {'p': 0.2880407124681934, 'r': 0.3419939577039275, 'f': 0.31270718232044203},** • **'PER': {'p': 0.7667120799273388, 'r': 0.6494960375471263, 'f': 0.7032532178114718},** • **'LOC': {'p': 0.5254826926421986, 'r': 0.5127871449586964, 'f': 0.5190573007645829},**

• **'MISC': {'p': 0.33263678621689624, 'r': 0.41967071989669646, 'f': 0.37111930534078746}** • 541.4375s	• **'MISC': {'p': 0.29470470823059763, 'r': 0.4073352807530023, 'f': 0.34198514885210163}** • 24507.328125s
Nlp_en2, evaluation: nlp_en3 • **'ents_p': 0.9261398223734528, 'ents_r': 0.8877302173880519, 'ents_f': 0.9065283485860083,** • **'ents_per_type':** • **'DATE': {'p': 0.9509482007639215, 'r': 0.8853328342379437, 'f': 0.9169682088394934},** • **'PERSON': {'p': 0.9368882259718467, 'r': 0.9118954722801924, 'f': 0.9242229170498437},** • **'CARDINAL': {'p': 0.9207832665776591, 'r': 0.9025737967136833, 'f': 0.9115876046409164},** • **'MONEY': {'p': 0.9619619619619619, 'r': 0.9348249027237354, 'f': 0.9481993093241243},** • **'TIME': {'p': 0.8867212083071114, 'r': 0.8114801305432905, 'f': 0.8474338412189254},** • **'GPE': {'p': 0.9647013928639573, 'r': 0.9586651497914297, 'f': 0.9616737993342843},** • **'WORK_OF_ART': {'p': 0.6693121693121693, 'r': 0.5227272727272727, 'f': 0.5870069605568445},** • **'LAW': {'p': 0.5769230769230769, 'r': 0.5263157894736842, 'f': 0.5504587155963302},** • **'ORG': {'p': 0.7665355738291026, 'r': 0.6536585365853659, 'f': 0.7056113213756787},**	Nlp_en, evaluation: nlp_en3 • **'ents_p': 0.8896865615615616, 'ents_r': 0.8289444922108392, 'ents_f': 0.8582421147129159,** • **'ents_per_type':** • **'DATE': {'p': 0.8991212653778559, 'r': 0.876027397260274, 'f': 0.8874241110147442},** • **'PERSON': {'p': 0.9588131089459699, 'r': 0.7863005738359846, 'f': 0.8640300115736123},** • **'MONEY': {'p': 0.9379310344827586, 'r': 0.8568856885688569, 'f': 0.8955785512699906},** • **'GPE': {'p': 0.9646411982178406, 'r': 0.9495194550713819, 'f': 0.9570205962569359},** • **'TIME': {'p': 0.8622339153312605, 'r': 0.7650679117147708, 'f': 0.8107500281119983},** • **'WORK_OF_ART': {'p': 0.3010471204188482, 'r': 0.21821631878557876, 'f': 0.25302530253025307},** • **'LAW': {'p': 0.5901639344262295, 'r': 0.36363636363636365, 'f': 0.45},** • **'ORG': {'p': 0.5108883511925337, 'r': 0.6786042240587695, 'f': 0.5829225004929994},** • **'PRODUCT': {'p': 0.2558139534883721, 'r': 0.1724137931034483, 'f': 0.2059925093632959},**

• **'LANGUAGE': {'p': 0.959349593495935, 'r': 0.9053708439897699, 'f': 0.9315789473684211},** • **'NORP': {'p': 0.8994502977553825, 'r': 0.9248704663212435, 'f': 0.9119832791453785},** • **'ORDINAL': {'p': 0.9833333333333333, 'r': 0.9681200187529302, 'f': 0.9756673753838884},** • **'FAC': {'p': 0.7085714285714285, 'r': 0.7774294670846394, 'f': 0.741405082212257},** • **'QUANTITY': {'p': 0.9097978227060654, 'r': 0.9, 'f': 0.9048723897911833},** • **'PERCENT': {'p': 0.9352750809061489, 'r': 0.9897260273972602, 'f': 0.961730449251248},** • **'LOC': {'p': 0.920120572720422, 'r': 0.8622881355932204, 'f': 0.8902661319722932},** • **'EVENT': {'p': 0.8645533141210374, 'r': 0.7957559681697612, 'f': 0.8287292817679559},** • **'PRODUCT': {'p': 0.26704545454545453, 'r': 0.34558823529411764, 'f': 0.3012820512820513}** • 740.21875s	• **'CARDINAL': {'p': 0.8952491646084556, 'r': 0.8119646857293451, 'f': 0.851575456053068},** • **'LANGUAGE': {'p': 0.9794174757281553, 'r': 0.7435141509433962, 'f': 0.8453159041394336},** • **'NORP': {'p': 0.7782956058588548, 'r': 0.8901015228426395, 'f': 0.8304522851053753},** • **'ORDINAL': {'p': 0.9741298212605832, 'r': 0.9530602853198343, 'f': 0.9634798790416376},** • **'FAC': {'p': 0.7319034852546917, 'r': 0.48576512455516013, 'f': 0.5839572192513368},** • **'QUANTITY': {'p': 0.8652373660030628, 'r': 0.786908077994429, 'f': 0.8242159008023341},** • **'PERCENT': {'p': 0.8938906752411575, 'r': 0.9174917491749175, 'f': 0.9055374592833876},** • **'LOC': {'p': 0.8869760479041916, 'r': 0.8183701657458563, 'f': 0.8512931034482758},** • **'EVENT': {'p': 0.7955801104972375, 'r': 0.6180257510729614, 'f': 0.6956521739130435}** • 23592.46875s
Nlp_multi, evaluation: nlp_en3 • **'ents_p': 0.027366633649356164, 'ents_r': 0.023741130834008493, 'ents_f': 0.025425288565193386,** • **'ents_per_type':** • **'DATE': {'p': 0.0, 'r': 0.0, 'f': 0.0},**	Nlp_en, evaluation: nlp_multi • **'ents_p': 0.03122573043035805, 'ents_r': 0.016971849436697623, 'ents_f': 0.021991079090557607,** • **'ents_per_type':** • **'PER': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'PERSON': {'p': 0.0, 'r': 0.0, 'f': 0.0},**

• **'PERSON': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'PER': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'LOC': {'p': 0.48740225890529976, 'r': 0.04156787196206283, 'f': 0.07660271727998907},** • **'MISC': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'GPE': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'WORK_OF_ART': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'LAW': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'LANGUAGE': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'NORP': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'ORG': {'p': 0.1657142857142857, 'r': 0.20525783619817997, 'f': 0.18337850045167117},** • **'ORDINAL': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'TIME': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'FAC': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'CARDINAL': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'EVENT': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'PRODUCT': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'QUANTITY': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'MONEY': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **‘PERCENT’: {‘p’: 0.0, ‘r’: 0.0, ‘f’: 0.0}** • 357.875s	• **'DATE': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'LOC': {'p': 0.046473906911142455, 'r': 0.4551104972375691, 'f': 0.0843358075249552},** • **'MONEY': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'GPE': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'TIME': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'ORG': {'p': 0.3202779532533165, 'r': 0.23256880733944954, 'f': 0.2694658517140579},** • **'PRODUCT': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'MISC': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'LANGUAGE': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'NORP': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'CARDINAL': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'ORDINAL': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'WORK_OF_ART': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'QUANTITY': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'PERCENT': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'EVENT': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'FAC': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'LAW': {'p': 0.0, 'r': 0.0, 'f': 0.0}** • 23334.15625s
Nlp_nl2, evaluation: nlp_nl3 • **'ents_p': 0.8606562201999035, 'ents_r': 0.7921140739617081, 'ents_f': 0.8249638891396587,** • **'ents_per_type':**	Nlp_nl, evaluation: nlp_nl3 • **'ents_p': 0.008297824624355236, 'ents_r': 0.008156367110000735, 'ents_f': 0.008226487808493294,** • **'ents_per_type':** • **'GPE': {'p': 0.0, 'r': 0.0, 'f': 0.0},**

• **'GPE': {'p': 0.9224840669927376, 'r': 0.9002675923916974, 'f': 0.9112404377585009},** • **'DATE': {'p': 0.873966258683427, 'r': 0.8155579564747646, 'f': 0.84375249500998},** • **'ORG': {'p': 0.49884704073789393, 'r': 0.2930022573363431, 'f': 0.3691695108077361},** • **'PERSON': {'p': 0.9278590078328982, 'r': 0.8380774945168974, 'f': 0.8806859720208667},** • **'WORK_OF_ART': {'p': 0.2882442748091603, 'r': 0.2821279139270771, 'f': 0.2851533001057242},** • **'CARDINAL': {'p': 0.9023036825290062, 'r': 0.9237390256498537, 'f': 0.9128955427015991},** • **'NORP': {'p': 0.7579499323410014, 'r': 0.6610119486649948, 'f': 0.7061697265778898},** • **'LANGUAGE': {'p': 0.8083994985374008, 'r': 0.7608652900688299, 'f': 0.7839124708742783},** • **'TIME': {'p': 0.8921953958450309, 'r': 0.8456625864821714, 'f': 0.8683060109289619},** • **'ORDINAL': {'p': 0.9514632405424697, 'r': 0.9315164220824598, 'f': 0.9413841807909605},** • **'FAC': {'p': 0.38613861386138615, 'r': 0.2805755395683453, 'f': 0.325},** • **'EVENT': {'p': 0.6444833625218914, 'r':**	• **'LOC': {'p': 0.8006872852233677, 'r': 0.0165014164305949, 'f': 0.03233640968704462},** • **'DATE': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'PER': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'PERSON': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'CARDINAL': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'MISC': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'LANGUAGE': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'TIME': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'EVENT': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'NORP': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'WORK_OF_ART': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'ORG': {'p': 0.3302564102564103, 'r': 0.24617737003058104, 'f': 0.2820849759088918},** • **'LAW': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'FAC': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'QUANTITY': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'ORDINAL': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'PRODUCT': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'PERCENT': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'MONEY': {'p': 0.0, 'r': 0.0, 'f': 0.0}** • 18424.796875s

0.5954692556634305, 'f': 0.6190075693860388}, • **'LOC': {'p': 0.509375, 'r': 0.4527777777777778, 'f': 0.4794117647058823},** • **'MONEY': {'p': 0.5661375661375662, 'r': 0.34627831715210355, 'f': 0.42971887550200805},** • **'PRODUCT': {'p': 0.05825242718446602, 'r': 0.06315789473684211, 'f': 0.06060606060606061},** • **'LAW': {'p': 0.1125, 'r': 0.08571428571428572, 'f': 0.0972972972972973},** • **'QUANTITY': {'p': 0.7335092348284961, 'r': 0.7830985915492957, 'f': 0.7574931880108993},** • **‘PERCENT’: {‘p’: 0.8056537102473498, ‘r’: 0.76, ‘f’: 0.7821612349914238}** • 807.875s	
Nlp_multi, evaluation: nlp_nl3 • **'ents_p': 0.005117444598756114, 'ents_r': 0.0046507161279701476, 'ents_f': 0.004872930082796688,** • **'ents_per_type':** • **'GPE': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'LOC': {'p': 0.4794520547945205, 'r': 0.006053268765133172, 'f': 0.011955593509820665},** • **'DATE': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'PER': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'CARDINAL': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'PERSON': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'LANGUAGE': {'p': 0.0, 'r': 0.0, 'f': 0.0},**	Nlp_nl, evaluation: nlp_multi • **'ents_p': 0.6934851596042562, 'ents_r': 0.6531198265379375, 'ents_f': 0.672697504923005,** • **'ents_per_type':** • **'LOC': {'p': 0.45933392627730013, 'r': 0.658356940509915, 'f': 0.5411257931195063},** • **'PER': {'p': 0.8547093894093806, 'r': 0.7236603257490986, 'f': 0.7837444791554455},** • **'MISC': {'p': 0.638135685505826, 'r': 0.4646368537900412, 'f': 0.5377380133058042},** • **'ORG': {'p': 0.3955773955773956, 'r': 0.24580152671755726, 'f': 0.3032015065913371}}**

• **'MISC': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'TIME': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'EVENT': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'NORP': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'WORK_OF_ART': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'ORG': {'p': 0.1991991991991992, 'r': 0.2126068376068376, 'f': 0.20568475452196386},** • **'LAW': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'FAC': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'QUANTITY': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'ORDINAL': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'PRODUCT': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'PERCENT': {'p': 0.0, 'r': 0.0, 'f': 0.0},** • **'MONEY': {'p': 0.0, 'r': 0.0, 'f': 0.0}** • 431.6875s	• 18476.90625s

Eigenständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, habe ich in jedem einzelnen Fall unter Angabe der Quelle kenntlich gemacht. Dies gilt auch für verwendete Zeichnungen, Skizzen, Ton- und Videoaufnahmen sowie graphische Darstellungen. Ich erkläre mich damit einverstanden, dass meine Arbeit im Verdachtsfall mithilfe einer Plagiatssoftware überprüft wird.

[illegible]

Düsseldorf, 06.10.2022