Machine Learning for NLP - NER report

Goya van Boven 2743245

November-December 2021

1 Introduction

This report focuses on the task of Named Entity Recognition and Classification (NERC). NERC refers to the automated recognition of *Named Entities*: any real-world "objects" that are referred to by a proper name. This could be a company, a geographical location, a person or a movie title (Jurafsky & Martin, 2014). Named entities can be divided into various categories, and recognizing such categories can be useful for further tasks, such as identifying customer sentiment towards companies in a specific sector through social media analysis.

In this study, I train various machine learning models on the CoNLL-2003 Named Entity dataset (Sang & De Meulder, 2003) and evaluate their performances against gold labels. Specifically, I train three feature-engineering models (naive Bayes, logistic regression and Support Vector Machines (SVM)) and two more advanced models (Conditional Random Fields (CRF) and BERT). For the feature-engineering models I experiment with including different feature combinations. I also evaluate two pre-trained models, namely Stanford CoreNLP (Manning et al., 2014) and Spacy (Honnibal & Johnson, 2015). I compare the performances of these different models, and carry out an ablation study to see which features contribute most to the feature-engineering models.

This report is structured as follows. Related wok on NERC is discussed in Section 2. The data is introduced in Section 3, and the pre-processing of the data is described in Section 4. Next, I describe the method (Section 5) and present the results (Section 6). The ablation study can be found in Section 7 and finally common errors are analysed in Section 8.

2 Related work

The term Named Entity Recognition and Classification was first coined at the Sixth Message Understanding Conference in 1996. The earliest NERC systems were based on manually created rules, ontologies and heuristics (Nadeau & Sekine, 2007; Rau, 1991). Later systems were mostly based on machine learning and feature-engineering, using models such as Hidden Markov Models (Zhou & Su, 2002; Malouf, 2002), Decision Trees (Carreras et al., 2002), Support Vector Machines (Y. Li et al., 2004) and Conditional Random Fields (Liu et al., 2015). Features that were often included in these models were (i) word level features such as capitalization, punctuation, morphology and Part-of-speech (POS) tags, (ii) list lookup features or gazetteers, i.e. lists of words that include common Named Entities, and (iii) corpus features such as the position in the sentence or paragraph, co-occurrences and the word and phrase frequency (Nadeau & Sekine, 2007).

Collobert et al. (2011) introduced the first neural network NERC system with minimal feature-engineering, after which such systems quickly gained popularity. In this system handcrafted feature vectors are replaced with word embeddings. The benefit of such models is that they are less domain dependent, since they do not require resources such as lexicons or ontologies which are specific to the domain (Yadav & Bethard, 2019). Neural approaches that have been used for NERC include achitectures based on RNN, biLSTM and BERT models (J. Li et al., 2020).

J. Li et al. (2020) discuss several open challenges in NERC. Firstly, training NERC models requires large training datasets which must be annotated, which is an expensive and time consuming task. Especially for low-resource languages and specialized domains the amount of required data remains a challenge. Continuing, the quality and consistency of these annotations are a concern. Between datasets, the labels assigned to the same entities can differ, which can cause problems if we use a model on data from another dataset than the one it was trained on. Secondly, model performances overall are much better on formal data than on informal user-generated data. This type of data is more difficult because utterances are often shorter and contain more noise. Finally, it remains difficult to recognize previously-unseen entities.

3 Data

The CoNLL-2003 dataset was originally introduced for the CoNLL-2003 Shared Task (Sang & De Meulder, 2003). This task focused on language-independent NER, where the data contained both English and in German sentences. The data for the two languages were presented in separate files, and this study uses the English part of the dataset. This English components takes its sentences from the Reuters Corpus (Lewis et al., 2004), which contains news articles from Reuters. The annotation of the data was done manually at the University of Antwerp, following MUC conventions (Chinchor et al., 1999). Following the IOB tagging scheme (Ramshaw & Marcus, 1999) four types of named entities are distinguished in the gold annotations:

- organisations (ORG),
- locations (LOC),
- persons (PER),
- miscelaneous names (MISC), containing all named entities that do not fit into any of the other categories.

Additionally, there is the label O that is assigned to words that are not identified as named entities. Finally, in the IOB tagging scheme all labels (except O) are preceded by either I- or B-, in order to distinguish between two entities of the same class that immediately follow each other: B indicates the beginning of a named entity and I is used for tokens inside of an entity.

Figure 1 shows the distribution of NERC labels in the gold data. This figure clearly shows that most words in the dataset (46,009) do not refer to a named entity. Of the named entities in the gold data, the B-PER label is most common (1,842) and I-LOC is the least common (257).

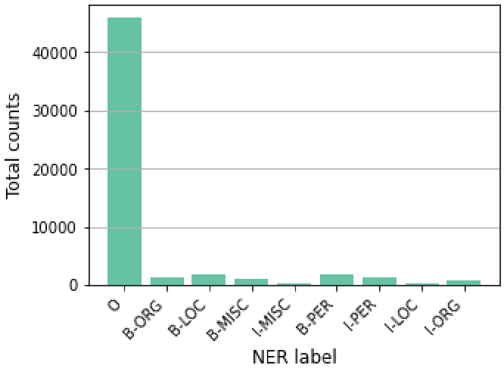

Figure 1: Distribution of NERC labels in the gold data

4 Data preprocessing

The Spacy and Stanford CoreNLP NERC model both use their own NERC labels. Therefore, I had to convert their labels to match the annotation scheme of the gold labels. In setting up the conversion rules I followed the tagging scheme of the CoNNL-2003 dataset. This means that all entities referring to organisations, locations and persons are mapped to their own label (ORG, LOC and PER respectively), and all other categories are converted to MISC. Notably, the annotation schemes of both models were much richer than that of the CoNLL-2003 dataset, containing more fine grained information on Named Entity types. For instance, the Stanford CoreNLP model recognizes the sepreate labels DURATION, CRIMINAL_CHARGE, RELIGION and EMAIL, which were now all mapped to the MISC label. Therefore much detail was lost by preprocessing the NERC labels. See Table 1 for a full overview of NERC label conversions.

Algorithm	**Original label**	**New label**
Spacy	PERSON	PER
	ORG	ORG
	GPE, LOC	LOC
	DATE, CARDINAL, TIME, FAC, ORDINAL, LANGUAGE, LAW, MONEY, EVENT, PERCENT, PRODUCT, QUANTITY, WORK_OF_ART, NORP	MISC
	_	O
Stanford CoreNLP	PERSON	I-PER
	ORGANIZATION	I-ORG
	CITY, LOCATION, COUNTRY, STATE_OR_PROVINCE	I-LOC
	DATE, NATIONALITY, NUMBER, DURATION TIME, ORDINAL, TITLE, MISC, MONEY, PERCENT, SET, EMAIL CAUSE_OF_DEATH, IDEOLOGY, CRIMINAL_CHARGE, RELIGION	I-MISC

Table 1: Conversion rules used for preprocessing the NERC prediction labels of the pre-trained Spacy (top) and Stanford CoreNLP (bottom) models.

Another thing that has to be noted is that the Stanford model does not identify whether a token starts a new entity (B- prefix) or is inside of an entity (I- prefix). For this reason, I decided to assign all entities the B- prefix. This will influence the results for this the model as all I- entities will be wrongly predicted. But on the other hand, manually adding the correct B- or I- prefix would unjustly enhance the prediction scores of this model, as it simply does not provide this information.

5 Methods

5.1 Features

I create 3 settings for the features of models to be trained on. In the *basic* setting the only feature included is the token, which is one-hot encoded. The *extended* setting includes the following set of features:

- The token itself; one-hot encoded
- Capitalization of the token, a categorical feature that contains the following categories: first letter capitalized, entire word capitalized, mixed capitalization (e.g. eBay) or no capitalization; one-hot encoded
- Whether the token contains numbers; binary
- Whteher the token contains punctuation[1]; binary
- The POS tag; one-hot encoded
- The previous token; one-hot encoded
- The POS tag of the previous token; one-hot encoded

Finally, in the *embeddings* setting, the data contains all features included in the *extended* setting except the previous token. But in this setting the token itself is represented by pre-trained Word2Vec word embeddings that were trained on the Google News dataset (Mikolov et al., 2013).

I expect that the *extended* setting is an improvement over the *basic* setting, as the latter setting includes more information. Further I also expect the *embeddings* setting to yield better results than the *extended* setting, as dense word embeddings contain more information than one-hot encoded vectors, even though their dimensionality is lower. For instance, two distinct semantically similar words will have similar word embeddings, while their one-hot encodings will be orthogonal.

5.2 Models

In total I evaluate 7 machine learning models, among which are 2 pre-trained models, 3 feature-engineering models and 2 more advanced models, which I briefly describe below.

Pre-trained models I evaluate 2 models that are pre-trained which I do not further fine-tune: Stanford CoreNLP (Manning et al., 2014) and Spacy (Honnibal & Johnson, 2015). The output labels of these models are preprocessed as described in Section 4.

Feature-engineering models These models include a logistic regression model, a Support Vector Machine (SVM) and a naive Bayes classifier. The first two models are trained on all of the 3 feature settings (*basic*, *extended* and *embeddings*). A linear kernel is used for the SVM, as the the data has a high dimensionality. For the naive Bayes classifier I specifically use the complement naive Bayes algorithm, because this algorithm is well suited for imbalanced datasets and often outperforms other naive Bayes implementations on text classification tasks (Rennie et al., 2003). I train this model on the *basic* and *extended* feature settings. I do not train it using the *embeddings* setting because word embeddings contain negative values, and this model cannot deal with negative values.

Advanced models Finally I train two more advanced models: a Conditional Random Field (CRF) model and a BERT model (Devlin et al., 2018). The CRF model is trained with gradient descent using the L-BFGS method, allowing for all possible transitions, with maximum iterations = 100, L1 regularization = 0.1 and L2 regularization = 0.1. The pre-trained BERT base cased model was fine-tuned for 2 epochs using a learning rate of $1e^{-5}$ and a batch size of 4. The inputs for these advanced models are the sentences rather than tokens, as they take the sequence structure into account.

The BERT model was implemented using Transformers (Wolf et al., 2020). For implementing the other models Scikit-Learn (Pedregosa et al., 2011) was used.

I expect the State-of-the-Art BERT model to perform best. Further, I expect the naive Bayes model to perform worse than the other two feature engineering models, as this generative model assumes independence between the features, while this likely is not always the case. Finally I expect the performance of the Standford CoreNLP to be lower than for the other models, as the predictions of this pre-trained model did not include B- and I- prefixes (see Section 4).

6 Results

I measure performance through precision, recall and F1-score. For all these measures, I report the macro-average, i.e. the unweighted average of the F1-scores for each label. I prefer this metric over the micro-average because the distribution of the labels is heavily skewed (see Figure 1), while the most common label (O) is not more important than the other labels. The micro-average considers all labels equally.

An overview of the model performances can be found in Table 2. This table shows that the highest performance is achieved by the BERT model (F1 = 93.52). Of the feature-engineering models, the SVM extended model achieves the best results (F1 = 84.49). All feature-engineering models benefit from adding additional featutes to the *basic* setting, but it is remarkable that the performances descrease from switching from one-hot encodings for the tokens to word embeddings, as word embeddings contain more information than one-hot encodings. Possibly, this can be explained by the fact that the following token is not included in the *embeddings* setting.

[1] In this study I consider the following characters as puntuation : $\#\$\%\&\backslash'()*+,-./:;<=>?@[]\wedge\_`\{|\}\sim$

The lowest scores are obtained by the pre-trained Allen CoreNLP (F1 = 40.43). This is according to expectations, as this is likely due to the fact that the output labels of this model do not include I- and B- prefixes, and all entities were assigned B-prefixes during preprocessing, resulting in a zero score for all I-labels (see Section 4).

	Precision	**Recall**	**F1**
Spacy	53.86	62.77	57.97
Allen CoreNLP	34.33	49.16	40.43
Logistic Regression basic	81.13	54.76	65.38
Logisctic Regression extended	85.97	78.98	82.32
Logistic Regression embeddings	81.21	77.60	79.36
SVM basic	80.57	64.99	71.94
SVM extended	86.71	82.39	84.49
SVM embeddings	81.13	75.67	78.30
Naive Bayes basic	78.71	64.73	71.04
Naive Bayes extended	76.66	75.93	76.29
CRF	85.52	80.7	83.04
BERT	93.21	93.84	93.52

Table 2: Model performances for the considered models in terms of precision, recall and F1 score. All scores are macro-averaged.

7 Ablation study

In order to investigate which features specifically contributed to the high performance of the SVM *extended* model, I carry out an ablation study. I select this model because it achieves the best performance of all feature-engineering models. The results for the ablation study can be found in Table 3. This table shows that the one-hot encoded tokens are the most important feature (F1 -16.49), followed by the previous token (F1 -3.67). When both of these features are excluded, the drop in performance much larger than the sum of their marginal effects (F1 -42.47). After excluding both features, the POS tag of the previous token becomes the most important (an additional drop of 10.33 in F1), while this feature only had a small effect (F1 -0.91) when the token and the previous token were still included. After the token and the previous token are excluded, the current POS tag and capitalization also have a small effect (additional drop of 2.32 and 2.31 respectively), while number and punctuation still do not contribute, the F1 score even improves slightly by additionally excluding these features. The POS tag of the previous word thus appears to be more important than the POS tag of the current token.

Excluded feature	**Precision**	**Recall**	**F1**	**∆ F1**
All featues	86.71	82.39	84.49	–
– *Token (as one-hot)*	70.49	65.69	68.00	-16.49
– *POS tag*	86.16	82.39	84.23	-0.26
– *Capitalization*	86.61	81.43	83.94	-0.55
– *Number*	86.64	82.34	84.44	-0.05
– *Punctuation*	86.67	82.34	84.45	-0.04
– *Previous token*	84.47	77.47	80.82	-3.67
– *Previous POS*	85.73	81.53	83.58	-0.91
– *Token & Previous token*	43.78	44.26	42.02	-42.47
– *Token & Previous token & POS*	43.13	36.77	39.70	-44.79
– *Token & Previous token & Capitalization*	41.1	38.41	39.71	-44.78
– *Token & Previous token & Number*	44.19	43.61	43.90	-40.59
– *Token & Previous token & Punctuation*	44.00	43.60	43.80	-40.69
– *Token & Previous token & Previous POS*	38.10	27.13	31.69	-52.80

Table 3: Ablation study for the SVM embedding model. Model performance by removing one of the considered features at a time.

To investigate whether the same features are important for the logistic regression model, I also excluded the token and the previous token from this model. Excluding the token gives a drop in F1 of 14.79, excluding the previous token decreases the F1 by 4.20. The contributions of these features thus seem to be of a similar scale as for the SVM.

8 Error analysis

For the error analysis, I will again focus on the best performing model, the SVM *extended* model. Table 4 shows the performance for each label by this model. The highest scores are obtained for the O and I-PER label, while the lowest performance is for the I-MISC, I-ORG and I-LOC labels, which all have an F1-score below 78. For I-MISC and I-ORG this is mostly due to a low recall score (65.0 and 68.3 respectively). Notably, these are also the three least common labels in the dataset (Figure 1).

Table 5 shows the confusion matrix for this model. This table shows that the following mistakes are common:

- B-ORG → B-LOC (100), B-PER (100)
- I-ORG → O (79), I-PER (63) , I-LOC (38), B-LOC (23)
- I-MISC → O (41), I-PER (26)
- I-LOC → I-ORG (18), I-PER (17), O(12)

I will further investigate the three most common of these mistakes: B-ORG → B-LOC, B-ORG → B-PER and I-ORG → O. Note however, that as some labels are generally less common, there are likely to be less absolute mistakes for these classes, while there can be a higher relative number of mistakes. I decided nonetheless to analyse the errors with the highest absolute occurrences to be sure to have a large selection of samples to analyse, in order to be able to find patterns that caused or influence the mistakes. But these are thus not necessarily the most important or harmful mistakes. Following (Wu et al., 2019), I aim to formulate precise hypotheses about the causes of errors, analyse all relevant instances (rather than only a selection of samples) and test the error hypotheses explicitly.

	Precision	Recall	F-score
O	99.1	99.6	99.3
B-ORG	82.6	77.6	80.0
B-LOC	87.6	87.3	87.4
B-MISC	87.8	79.8	83.6
I-MISC	85.6	65.0	73.9
B-PER	87.5	90.4	88.9
I-PER	88.5	95.7	92.0
I-LOC	78.1	77.8	77.9
I-ORG	83.6	68.3	75.2

Table 4: Performance per label for the SVM *extended* model

	O	B-ORG	B-LOC	B-MISC	I-MISC	B-PER	I-PER	I-LOC	I-ORG
O	42606	26	12	21	14	38	9	2	31
B-ORG	49	1041	100	24	1	100	4	1	21
B-LOC	59	87	1604	23	0	46	10	4	4
B-MISC	54	55	40	736	5	25	2	0	5
I-MISC	41	13	4	13	225	5	26	8	11
B-PER	58	28	41	12	0	1665	32	0	6
I-PER	25	2	3	2	1	15	1251	3	5
I-LOC	12	2	5	0	2	1	17	200	18
I-ORG	79	6	23	7	15	7	63	38	513

Table 5: Confusion matrix for the SVM *extended* model

8.1 B-ORG → B-LOC

Here, I will investgate the 100 instances where B-ORG was the true label, but B-LOC was predicted. These 100 errors occur for 68 unique tokens. For 23 of these tokens, the model sometimes makes correct classifications, totally creating 66 correct predictions for these tokens. Many of the words for which this error occurs are names of locations, such as *China, Florida* and *Bucharest.* But in these specific sentences, the location name is (i) part of the name of a company or organisation (e.g. *London Newsroom*, *New York Yankees*) or (ii) is used to refer to a sports team (as in *Baratelli , who played for Nice and Paris St Germain , takes over from Albert Emon*). This raises the hypothesis that these words are misclassified because they usually refer to locations, rather than to organisation. Concretely, this would mean that these words have more B-LOC annotated occurrences in the gold data, than B-ORG annotations. To see whether this hypothesis holds, I compare the annotations in the data. I find that 38.24% of the 68 words for which this error occurs also have the B-LOC label in the dataset, and that for 20.59% the B-LOC label is more common. The hypothesis may thus hold for one fifth of the words, but for the remaining ones this explanation does not suffice.

Continuing, I note that all words for which this error occurs either have a capitalized first character or the entire word is capitalized. Capitalization might thus play a role here. The ablation study (Section 7) shows that capitalization has little impact on the predictions, but it might just be these couple of instances for which it is important. Therefore, I compare the distributions of capitalization values for the labels B-LOC and B-ORG. As can be seen in Table 6 the capitalization value distributions of these two labels is very similar. Capitalization is therefore unlikely to influence confusions between these labels.

Label	**All**	**First**	**Mixed**	**No Capitalization**
B-ORG	23.0%	75.7%	1.3%	0.1%
B-LOC	23.6%	75.5%	0.8%	0.1%

Table 6: Distribution of capitalization values of words with the labels B-ORG and B-LOC

A final thing I note for these mistakes is that they often from an entity with a word that follows it, such as *London Newsroom*. But currently, the model only has access to the preceding and not to the following word. So for the model, a regular instance of *London* and the combination *London Newsroom* are identical. Therefore, I think it would be beneficial to include the next token as well. This way the model can also perceive the difference between these instances, and its performance is likely to improve.

8.2 B-ORG → B-PER

There are 100 tokens with a B-ORG annotations that are predicted to be B-PER entities. These errors are made for 49 unique words, which are mostly names that usually refer to people, such as *Jones, Douglas, Lola* and *Johnson*. But in these specific contexts these names refer to companies or organisations (as for instance *Douglas & Lomason*). Interestingly however, only 1 of the words for which the model makes this type of error also has the B-PER label in the dataset, and even for this word (*Johnson*) the B-ORG label is still more common. For 11 of the 49 words the model also sometimes makes correct predictions for the word having the B-ORG label, totally creating 26 correct classifications.

To get more insights into what causes these mistakes, I scanned the sentences in which these tokens occur, and got the idea that the preceding POS tag might influence the mistakes. The ablation study (Section 7) confirms that this feature carries some importance. To investigate this idea quantitatively, I compare the preceding POS tags for these error words with the overall preceding POS tag distributions for the B-ORG and P-PER label. As can be found in Table 7, the two most common preceding POS tags among these error words (NNP and NNP) more often precede B-PER than B-ORG entities. The preceding POS tag might thus have influenced the misclassifications. This is further confirmed by the observation that among the correct classifications for these error words, the preceding POS tag `IN` is the most common, which more prominently precedes B-ORG entities (8.28%) than B-PER entities (5.48%). Still, Table 7 also shows this hypothesis cannot explain all errors, as some of the common preceding tags still are more common for B-ORG entities. Altogether, it seems that the preceding POS tag might play some role in the misclassifications between these labels, but it cannot explain all mistakes.

Previous POS-tag	**Count**	**% preceding B-ORG**	**% preceding B-PER**
`NNP`	22	5.4 %	10.3%
`NN`	18	2.9 %	9.8 %
`)`	11	1.6 %	0.3 %
`NNS`	7	0.5 %	0.2 %
`VBN`	6	0.9 %	0.6 %
`.`	6	7.1 &	11.4 %

Table 7: The six most common preceding POS tags of B-ORG annotated tokens which are predicted to be B-PER, with the percentage of how often these POS tags precede the B-ORG and B-POS labels.

8.3 I-ORG → O

For this type of error, entities of the type I-ORG are not recognized to be named entities. In total there are 79 such mistakes, for 33 unique words. Different than for the mistakes discussed above, 66.67% of these words also have O annotations, and for 54.55% of these words the O label is more common. Looking at the words for which these mistakes are made, this seems sensible: these include words such as *the, of, group, trade, women* and *Sunday*. These words usually do not refer to named entities, but are part here of an n-gram that forms a named entity, such as for instance *the Test and County Cricket Board* or *the Chicago Board of Trade*. It is rather logical that these words are difficult to classify, especially considering the fact that the model only has access to the token that directly precedes it.

Still, 9 of the error words are also classified correctly, which in total add up to 42 correct predictions. These 9 words are : *for, and, in, of, s, (,), Financial* and *Group*. The word *of* alone has 18 correct classifications, compared to 7 incorrect predictions. As the word itself is the same between the correct and incorrect classifications, the only difference in the information the model has access to is the previous token and its POS tag. An example of a correct prediction is *Bank of* while *League of* is incorrectly classified. The exact influence of these preceding tokens should be more thoroughly analysed before any conclusions can be drawn, but I suspect that they play a role in these mistakes.

9 Discussion

In this study I compared various machine learning techniques on a NERC task. According to expectations, the best results were found for the State-of-the-Art BERT model. Of the feature-engineering models, the best results were for the SVM *extended* model. An unexpected finding is that replacing the one-hot token encoding with word embeddings degraded the performance of the feature-engineering models. This is against expectations, as word embeddings carry much more information than one-hot encodings. There could be two reasons why this setting yields lower results. Firstly the previous token was excluded in this setting, which is the second most important feature according to the ablation study (Section 7). Therefore it is likely that including the previous token as a dense word embedding would improve the models. Secondly, a linear kernel was used for the SVM model in all settings, while a Gaussian kernel would likely have been better in the *embeddings* setting, as the dimensionality of word embeddings is much lower, and Gaussian kernels are more suited for lower dimensional data.

Another more general limitation of this study is that I did not fine tune hyperparameters extensively, so better results can likely still be found with the same methods and data. A final limitation which is also mentioned in Section 8 is that currently no information about the following token was included in the feature sets. I believe the feature-engineering models could benefit from including information such as the token and the POS tag. For instance in n-grams such as *the New York Yankees*, knowing that the word *Yankees* follows *York* could provide clear information that the latter is an organisation and not a location, something that currently is not always predicted correctly.

In future work it would be interesting to address the limitations described above, and compare whether the models can be further improved. Moreover, it would be interesting to train a similar set of models on the German version of the CoNLL-2003 dataset, and compare whether the models perform similarly, whether the same features are important and if the same types of errors are made.

10 Conclusion

In this study, a variety of machine learning models were trained on the CoNLL-2003 dataset in order classify named entities. The models included two pre-trained models (Spacy and Allen CoreNLP), three feature-engineering models (logistic regression, SVM and Naive Bayes) and two more advanced models (CRF and BERT). The feature-engineering models were all trained in two settings: a *basic* setting where only the token was included and an *extended* setting where a broad variety of features was included. The logistic regression and the SVM model were also trained on a third *embeddings* setting, in which the tokens were represented by dense word embeddings. The best results were found for the BERT model, and the best feature-engineering model was the SVM in the *extended setting.* The results for the pre-trained models were less good, probably because the output-labels for these models did not match labels in the dataset directly, and had to be preprocessed first (see Section 4). I carried out an ablation study on the SVM *extended* model to get more insights into the contributions of the features, and found the token itself and the previous token to be the most important features. Finally, I carried out an error analysis on the most prominent confusions between classes. In future work it would be interesting to still include a broader selection of features and to test the same models on the German part of the CoNLL-2003 dataset.

References

Carreras, X., Marquez, L., & Padró, L. (2002). Named entity extraction using adaboost. In *Coling-02: The 6th conference on natural language learning 2002 (conll-2002).*

Chinchor, N., Brown, E., Ferro, L., & Robinson, P. (1999). Named entity recognition task definition. *Mitre and SAIC.*

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research, 12*(ARTICLE), 2493–2537.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805.*

Honnibal, M., & Johnson, M. (2015, September). An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1373–1378). Lisbon, Portugal: Association for Computational Linguistics. Retrieved from `https://aclweb.org/anthology/D/D15/D15-1162`

Jurafsky, D., & Martin, J. H. (2014). Speech and language processing. vol. 3. *US: Prentice Hall.*

Lewis, D. D., Yang, Y., Russell-Rose, T., & Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research, 5*(Apr), 361–397.

Li, J., Sun, A., Han, J., & Li, C. (2020). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering.*

Li, Y., Bontcheva, K., & Cunningham, H. (2004). Svm based learning system for information extraction. In *International workshop on deterministic and statistical methods in machine learning* (pp. 319–339).

Liu, S., Tang, B., Chen, Q., & Wang, X. (2015). Effects of semantic features on machine learning-based drug name recognition systems: word embeddings vs. manually constructed dictionaries. *Information*, *6*(4), 848–865.

Malouf, R. (2002). Markov models for language-independent named entity recognition. In *Coling-02: The 6th conference on natural language learning 2002 (conll-2002).*

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations* (pp. 55–60).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, *30*(1), 3–26.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Ramshaw, L. A., & Marcus, M. P. (1999). Text chunking using transformation-based learning. In *Natural language processing using very large corpora* (pp. 157–176). Springer.

Rau, L. F. (1991). Extracting company names from text. In *Proceedings the seventh ieee conference on artificial intelligence application* (pp. 29–30).

Rennie, J. D., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the 20th international conference on machine learning (icml-03)* (pp. 616–623).

Sang, E. F., & De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... Rush, A. M. (2020, 10). Transformers: State-of-the-Art Natural Language Processing. In (pp. 38–45). Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/2020.emnlp-demos.6`

Wu, T., Ribeiro, M. T., Heer, J., & Weld, D. S. (2019). Errudite: Scalable, reproducible, and testable error analysis. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 747–763).

Yadav, V., & Bethard, S. (2019). A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470*.

Zhou, G., & Su, J. (2002). Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 473–480).