Data reduction in named entity recognition

Karsten Langeark

Layout: typeset by the author using LaTeX.
Cover illustration: Unknown artist

Data reduction in named entity recognition

by use of shallow learning methods

Karsten Langeark
12567418

Bachelor thesis
Credits: 18 EC

Bachelor *Kunstmatige Intelligentie*

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisor
Prof. Giovanni Colavizza

Institute for Humanities Research
Faculty of Humanities
University of Amsterdam
Turfdraagsterpad 9
1090 GN Amsterdam

Semester 1, 2022

Abstract

Due to the cost required to build large data sets and the general lack of data when working with a lot of non-English languages, many NLP tasks struggle with the availability of data. Named Entity Recognition (NER) is one of the tasks that struggles with data availability. The HIPE 2022 clef evaluation lab shared a NER task with the goal of building NER models that can be deployed in multiple different languages. To help with the HIPE 2022 task, this paper proposes two models that can learn NER with little data. The first model, which is a decision tree model, reaches its top performance with a small amount of data but it performs worse than the models created for HIPE 2020. In contrast, the second model, a CRF model, beats the best models of HIPE 2020 without using all data available but it does not reach its top performance before using all the data.

Contents

Chapter 1

Introduction

In 2022 the CLEF-HIPE-2022 evaluation lab proposed a NER task based on a dataset with historical data from five different languages. HIPE 2022 had two objectives.

1. assess and advance the development of **robust, adaptable and transferable** named entity processing systems across languages, time periods, document types, and annotation tag sets.
2. deal with **challenging historical material**, thereby supporting information extraction and text understanding of cultural heritage data.

(CLEF-HIPE, 2022)

NER is a task where an algorithm has to recognize entities, times and quantities. In the last couple of years NER has been dominated by deep learning models. This trend can also be seen when looking at the top performing models from HIPE 2020 (Ehrmann, Romanello, Flückiger, & Clematide, 2020). One of the problems deep learning faces is overfitting when learning on small data sets (Hestness et al., 2017). So when developing a transferable model, as tasked by HIPE 2022, it is important to look at the data requirements for the models that are developed.

While English enjoys a relative luxury in the amount of data present, not every language shares this luxury. A study done by w3techs shows that on the 15th of June 2022 the internet consisted of 62.2% of English web pages, while many languages are not present in more than 0.1% of web pages (W3Techs, 2022). Besides the lack of data there is also a high cost associated with the creation of large data sets. This shows that when building a model that is transferable to different languages, a model is needed that can learn with a limited amount of data.

In that regard, this paper will look at shallow learning models. Where a shallow learning model is defined as any model that is not deep, that is any model that does not make use of any hidden layers.

Because it is important for the development of transferable models to have a model that can learn with little data this paper asks the question: Can shallow learning methods learn Named Entity Recognition with little data available?

This will be answered by comparing the results of the shallow learning methods at different amounts of data to the results from HIPE 2020 and also by looking at which point increasing the amount of data does not improve the performance of the model. If the data needed to outperform the average results from HIPE 2020 or to stop learning is less than the full amount of data available it will be seen as learning with little data.

Chapter 2

related work

This chapter will focus on the different NER used in the past, which challenges NER methods face and lastly it will take a look at HIPE 2020.

2.1 Methods

For NER there are three different types of methods used. The first methods used were ruled-based systems, these rules were based on patterns found in the data. The main advantage of these systems was that they needed no data to train and were easy to understand but they needed a significant amount of expertise and time to be developed.

At the turn of the century when more data became available the ruled-based systems were being replaced by machine learning models. These models learn from manually selected features. Models used at that time included decision trees, support vector machines, maximum entropy models and linear chain conditional random fields (CRF). CRFs proved very successful because they could take neighbouring tokens into account.

When looking at shallow learning methods for NER the field is dominated by CRF methods. The Stanford CRF classifier was the most popular method and was used for many different historical data sets and languages. The F-scores of the shallow learning methods use were generally in the 60 - 70% range.

The latest models in NER are deep learning methods with their main advantage being the ability to learn the input representation themselves without needing someone to create the features for them. Deep learning methods have seen development in two fields for NER: one focused on the architecture and the ability to handle context efficiently, the other being able to use word embedding and language representations like BERT (Ehrmann, Hamdi, Pontes, Romanello, & Doucet, 2021).

2.2 Challenges

There are four challenges NER methods have to deal with. These challenges arise from the differences in historical documents and optical character recognition (OCR) noise when digitizing the documents, this causes large and sparse feature spaces and this problem is only made more difficult by the lack of data.

The first challenge is the historical variety space. The historical data used for NER generally spans a larger time gap, uses different types of documents all of a different nature and is often written in multiple languages. This causes a large variety between the different documents used. Al these different document types combined with the problems of domain shifts, that many NLP methods face, make building generalized NER methods for historical data a challenge. This challenge of variety in space is not made easier when taking looking at the needs of humanities research. Where normal NLP tasks generally focus on a specific domain, and thus do not face the problem of domain shifts, humanities and social sciences are focused on a whole spectrum of historical documents.

The next challenge arises from the noise created by automatic text acquisition. Text acquisition is done in two steps, first by recognition of characters and secondly by recognition of layout. In both stages noise can occur which in turn creates more variable data as mistakes in text acquisition must be taken into account when building a NER.

NER also faces the challenge of changes in languages as time progresses. Historical languages differ a lot from modern languages which in turn hurts the performance of NER systems. These differences can be variations in the spelling, differences in naming conventions but also a drift in context. These difference combined with the challenges mentioned above creates a large variable feature space which would be a challenge for any learning algorithm.

These above challenges are not made easier by the last challenge NER faces which is a lack of data. Most NER methods depend on labelled data but unfortunately not a lot of historical annotated data sets are available which makes the learning process all the more difficult (Ehrmann et al., 2021).

2.3 HIPE 2020

The dominance of deep learning methods is clear when looking at HIPE 2020. Of all models used, all but two were deep learning methods and many also made use of BERT word embeddings (Ehrmann et al., 2020). Besides being the dominant method to use they also proved to be the better models as almost all deep learning methods outperformed the shallow learning methods and the shallow learning methods were also only able to work with one of the three languages.

Chapter 3

Methods and data

In this section the data and methods used in this paper will be explained. There are two models chosen for the experiment. The first one is a decision tree model, this is a model that can emulate rule based systems which were used in the past and might prove a suitable method to quickly find rules present in data. The second model used is a CRF model, this model was the most popular model in the past. This model was also used during HIPE 2020 and it managed to compete with the deep learning models.

3.1 Data

The data set used for training was the HIPE 2020 data set which consisted of newspaper articles from Swiss, Luxembourgish and American newspapers in French, German and English. The data set contains roughly 10.000 named entities per language. The data sets consisted of both coarse- and fine-grained named entities, but the model was only trained on coarse-grained named entities. Coarse-grained NER exist of fewer, more general, classes than fine-grained NER. For example, both Holland and Amsterdam would get class LOC for coarse-grained NER whereas they would be classified as a region and town by fine-grained NER. Because more classes generally need more data to learn and this paper focuses on building models that do not need a lot of data it was decided to use coarse-grained named entities for the training of the models. Only literal named entities were used for training as the data set had more literal named entities labelled. Besides the HIPE 2020 data set the AJMC data set was used. This data set is a set of historical commentaries based on three 19th centuries commentaries in French, German and English on Sophocles tragedy the Ajax. This data set was used to check the validity of the CRF method to see whether the high result achieved on the HIPE 2020 data set could be replicated when trained on a completely different data set. It should also be noted that the AJMC data set was roughly one third the size of the HIPE 2020 data set.

3.2 Decision trees

A decision tree algorithm builds a decision tree based on a given scoring method. At each layer, it decides to either split on a feature that provides the highest information gain, figure 3.1, or create a leaf node. When all remaining data points on a branch have the same label or if there are fewer data points left than the minimum amount required the algorithm will create a leaf node. The class given to the leaf node is whichever class holds the majority at the current node.

$$IG_i = S_{i-1} - S_i$$

Figure 3.1: Formula for information gain. With IG_i being the information gain at split i and s_i the score at split i

The data is split into discrete and continuous features. A discrete feature will be split over all different values of said feature where a continuous feature will be split on one value where one side of the tree will be all data points with a value lower than the split value and the other side will contain all data points higher or equal to the split value. When predicting a class for a data point the decision tree will go through each node, according to the features of the data point, until it reaches a leaf node. This leaf node will be the class assigned to the data point. During the experiment two different scoring methods were tested, Entropy, figure 3.2, and Gini index, figure 3.3. Both had similar results but Entropy was faster than Gini index so it was decided to use Entropy.

$$E(S) = \sum_{i}^{c} -p_i log_2(p_i)$$

Figure 3.2: Formula for entropy. With $E(s)$ being the entropy at split S, c are all the classes in the current split and p_i is the probability of the a class

$$G(S) = 1 - \sum_{i}^{c} -p_i^2$$

Figure 3.3: Formula for gini index. With $G(s)$ being the gini index at split S, c are all the classes in the current split and p_i is the probability of the a class

3.2.1 Features

The paper *A survey of named entity recognition and classification* describes three types of features for NER (Nadeau & Sekine, 2007).

1. Word-level features
2. List lookup features
3. Document and corpus features

Word-level features describe the features that make up a token, such as whether the token contains numerical values or if it starts with a capital letter. They also describe the pre- and postfixes of tokens.

List lookup features are features that describe an "is a" relation in a language. For example, Germany is a country or red is an adjective. But they can also describe tokens that are frequently used in certain types of named entities, like how "associates" is generally used for companies or how the postfix "dam" is used in several Dutch town names.

Document and corpus features describe the features that are present in the document or corpus, for example the other tokens in a corpus but also the location of the current token.
Four different word-level features are used:

1. Whether the token is capitalised
2. Whether the token contains a number
3. The last two characters of the token
4. The last three characters of the token

The first two features are represented by a Boolean value while the latter two are categorical values where each sub-string is represented by a numerical value.

Because list lookup features require previous knowledge of a language they were ignored. This is done because this research aims to develop models that can be used for languages with little data available, so the assumption is made that these list lookup features will not be available for all languages. Therefore the model must be able to train without them.

There a three different Document and corpus features used:

1. The relative position of other tokens.

2. Whether the token is at the end of a sentence
3. The location of the token

For the first feature, the only tokens that are used are the ones that make up 50% of the corpus. If a token does not appear in a sentence it will be given infinite as a value. The second feature is described by a Boolean value. The last feature is the index of the token in a sentence.

3.3 CRF

CRF, conditional random fields, predict the class of a sample by looking at the features of a sample while also taking the neighbouring samples into account. The model used is the sklearn-crfsuite model (Korobo, 2017). Sklearn uses a linear chain CRF which predicts the probability of a class by only looking at the features of a token and the classes of its neighbours. To calculate the probability of a token being a certain class the algorithm would use to formula described in figure 3.4. To predict the named entities in a sentence the CRF would look for a sequence of classes which would yield the highest joint probability for the given sentence.

$$P(Y_i = c|X_i, Y_{i-1}, Y_{i+1})$$

Figure 3.4: Formula for calculating the probability of a token being a certain class, with Y being a token, i being the index of a token and X being the features of a token

3.3.1 Features

The following features are used for the CRF:

1. The lower case of the token
2. The last three characters of the token
3. The last two characters of the token
4. Whether the whole token is uppercase
5. Whether the token is capitalized
6. Whether the token contains a number
7. The lower case of the previous token

8. Whether the whole previous token is uppercase
9. Whether the previous token is capitalized
10. The lower case of the next token
11. Whether the whole next token is uppercase
12. Whether the next token is capitalized

The features of the token itself were chosen as they were the same as the features used by the IRISA team from HIPE 2020 (El Vaigh, Le Noé-Bienvenu, Gravier, & Sébillot, 2020). Although the labels of the previous and next token are already taken into account in this CRF, the tokens themselves are also taken into account to give a bit more context to the model. If the word is at the beginning of the sentence or the end of the sentence the token will get *BOS* or *EOS* value instead of a previous or next token.

Chapter 4

Results

The data was split into training and test sets. The model is trained on a training set and then tested on a test set. The performance of the model is judged on both recall and precision. For testing, five different test sets were created and each test set consisted of a random 10% of the data. Five different test sets were created to prevent bias in the result. As the test sets are chosen randomly some test sets will have better results than others as the similarity between the test set and the training set will differ. To minimize this bias multiple test sets are created so that outliers in the test sets will get balanced out by other test sets that do not contain these outliers. For every test set 90 random training sets were created for the model. 10 random training sets for every 10% of the total amount of data. Multiple training sets per slice of data were used to prevent a biased training set from affecting the results. By measuring performance at different slices of the data it is possible to see how much data a model needs to learn as, at a certain point there will not be any significant improvement when increasing the amount of data.

The models where trained on the English, French and German data from the HIPE 2020 data set. Because the CRF model reached doubtfully high results it was decided to train and test the model on the AJMC data set as well to see if similar results would be achieved.

4.1 Result decision tree

The precision of all models was 1 in every case and the mean recall of the models stayed stable in all data slices except for the French model which saw an increase between 10% and 20% but stayed stable afterwards. English reached a mean recall of 0.3, French a mean of recall of 0.4 and German of 0.22. The distribution of the recall did decrease when increasing the amount of data. As decision trees already struggled with predicting whether a token was a named entity it was decided to not have a decision tree try to

predict which class of named entity a token was.
The complete distribution of the recall is shown in figures 4.1, 4.2 and 4.3.

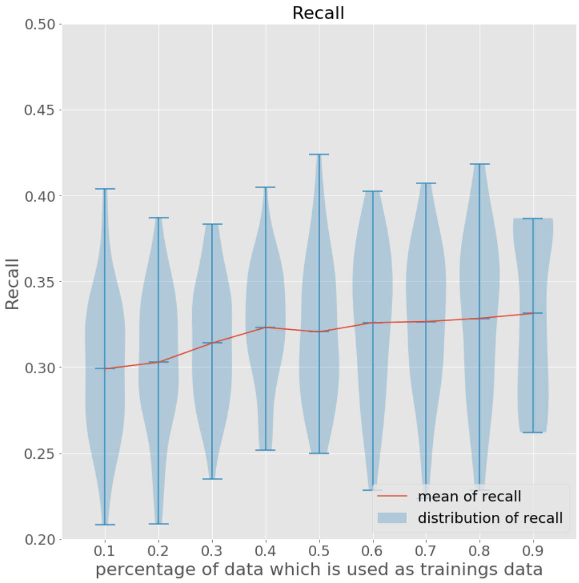

Figure 4.1: Results of the decision tree model on the English HIPE 2020 data

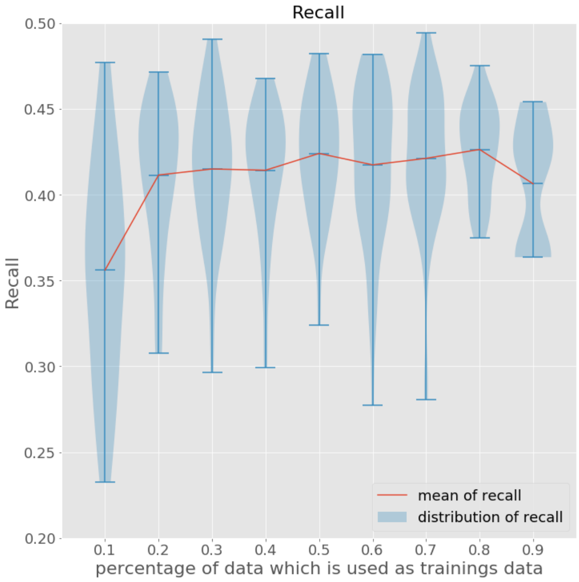

Figure 4.2: Results of the decision tree model on the French HIPE 2020 data

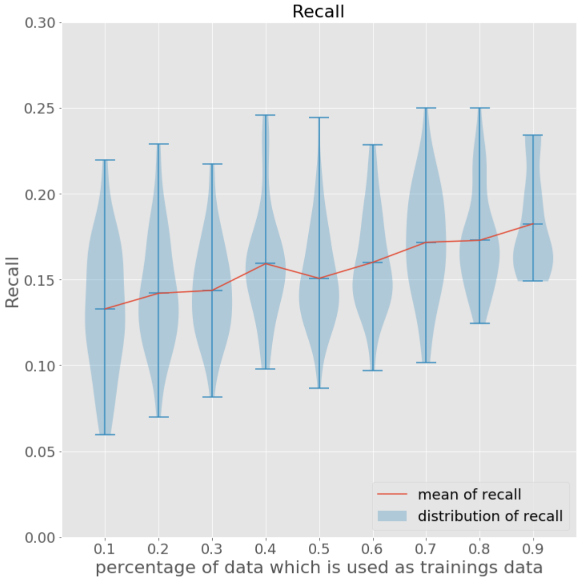

Figure 4.3: Results of the decision tree model on the German HIPE 2020 data

4.2 CRF

Unlike decision trees, there is an increase in both recall and precision when there is more data available. Where the precision of the three different languages is mostly similar, but the recall of the different languages does share some differences. German starts with the lower recall and also sees a bigger distribution whereas French has the highest recall to start with. When increasing the data the difference in recall between the different languages becomes less. When increasing the data to 90% the results of both the recall and precision converge to 1.
Because the relatively high results of the CRF when compared to the results of HIPE 2020 the CRF model was trained and tested on the AJMC data set to see if similar results would be achieved. When tested on this data the models learned even quicker with most the French and German models reaching a precision, recall and f1 score of 1 almost immediately and although the English data set shows a slight learning curve it also comes close to 1 when the maximum data is achieved.

The complete distribution of recall, precision and f1 for the HIPE 2020 data is shown in figures 4.4, 4.5 and 4.6. Table 4.1 shows the average result of the CRF model on the AJMC data set. As there was little distribution in the results the complete distribution of the three languages in shown in Appendix A.

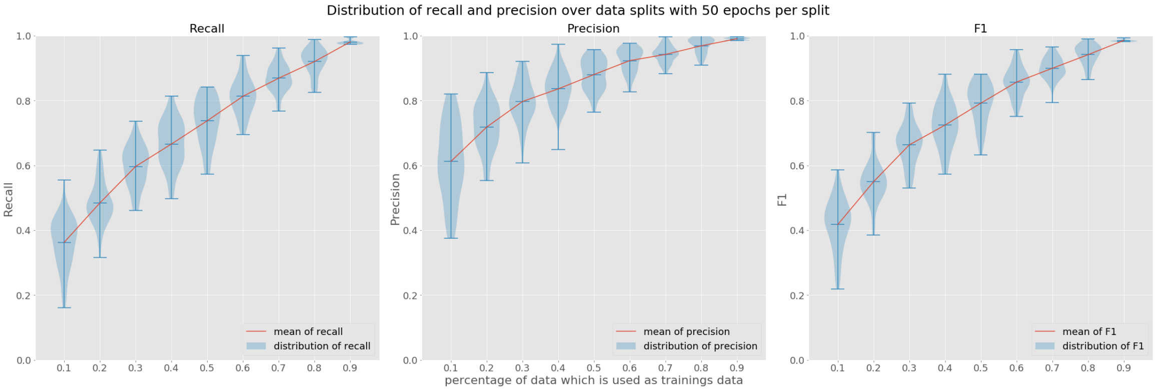

Figure 4.4: Results of the CRF model on the English HIPE 2020 data

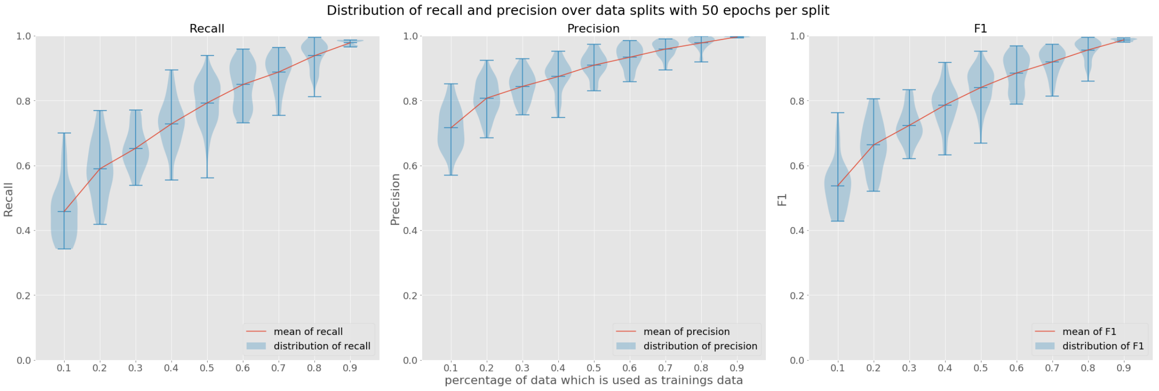

Figure 4.5: Results of the CRF model on the French HIPE 2020 data

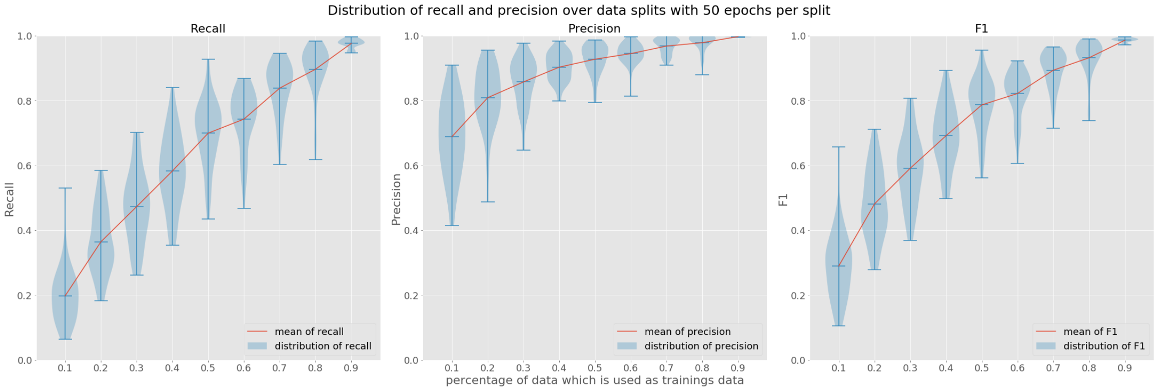

Figure 4.6: Results of the CRF model on the German HIPE 2020 data

	French			German			English		
Model name	P	R	F	P	R	F	P	R	F
CRF_0.1	.955	.994	.974	.956	.991	.973	.875	.751	.804
CRF_0.2	.971	.995	.983	.974	.993	.983	.899	.824	.856
CRF_0.3	.977	.995	.986	.980	.994	.987	.916	.876	.893
CRF_0.4	.983	.996	.989	.986	.995	.990	.932	.903	.916
CRF_0.5	.988	.997	.992	.987	.996	.992	.950	.929	.938
CRF_0.6	.991	.997	.994	.992	.997	.994	.961	.946	.953
CRF_0.7	.993	.998	.996	.995	.997	.996	.974	.967	.970
CRF_0.8	.996	.999	.998	.997	.998	.997	.981	.980	.980
CRF_0.9	.999	1.00	.999	.998	.999	.999	.990	.991	.990

Table 4.1: Average results of the CRF model on the AJMC data over each split.

4.3 HIPE 2020

In table 4.2 the results of HIPE 2020 are displayed. Only the results of fuzzy NER on literal coarse NER are displayed as those are also the criteria on which the other models are tested. With fuzzy NER the results only take if a single token is correctly classified instead of the entire named entity. For example with the name John Doe, if a strict measure is selected the entire name John Doe has to be correctly classified to count as a true positive while with fuzzy criteria a true positive will be added for both John and Doe separately if one of them is correctly classified. In table 4.3 the average performance of decision tree models and the CRF models, on the HIPE 2020 dataset, are given as a comparison. As the decision tree saw limited improvement, only the results of the split with the maximum amount of data is given. The decision tree model performs worse than almost all models developed for HIPE 2020 whereas the CRF outperforms even the best models without using all the data.

	French			German			English		
Model name	P	R	F	P	R	F	P	R	F
CISTERIA	-	-	-	**.880**	.683	.769	-	-	-
EHRMAMA	.839	.861	8.77	.814	.765	.789	.405	.633	.494
ERTIM	*.604*	*.344*	*.439*	-	-	-	.-	-	-
INRIA	.755	.842	.796	-	-	-	.568	.746	.645
IRISA	*.828*	*.744*	*.784*	-	-	-	-	-	-
L3I	**.912**	**.931**	**.921**	.870	**.886**	**.878**	.794	**.817**	**.806**
LIMSI	.887	.909	.898	-	-	-	-	-	-
NLP-UQAM	.828	.744	.784	-	-	-	-	-	-
SBB	.765	.689	.725	.730	.708	.719	.642	.572	.605
SINNER	.886	.902	.894	.775	.819	.796	-	-	-
UPB	.825	.817	.821	.788	.740	.763	.743	.592	.659
UVA-ILPS	.794	.869	.830	.689	.768	.726	.635	.728	.678
WEBIS	.876	.273	.416	.833	.405	.545	**.873**	.122	.215
Baseline	.825	.721	.769	.790	.464	.585	.736	.454	.562
Median	.828	.829	.808	.801	.752	.766	.642	.633	.645

Table 4.2: Results for HIPE 2020 fuzzy coarse named entity recognition (precision, recall, F-measure) and the average results of the proposed models when trained on the maximum amount of data. Bold font indicate highest and italics indicate a shallow learning model

	French			German			English		
Model name	P	R	F	P	R	F	P	R	F
Decision tree	1.00	.406	.577	1.00	.182	.308	1.00	.331	.496
CRF_0.1	.716	.458	.538	.689	.198	.290	.612	.363	.417
CRF_0.2	.807	.589	.663	*.809*	.364	.481	*.718*	.484	.549
CRF_0.3	*.843*	.653	.723	.857	.473	.591	.797	.597	*.663*
CRF_0.4	.874	.728	.786	**.903**	.583	.691	.836	*.666*	.724
CRF_0.5	.909	.793	*.840*	.927	.699	*.786*	**.880**	.737	.792
CRF_0.6	**.933**	*.849*	.885	.944	.743	.822	.923	.813	**.857**
CRF_0.7	.959	.888	.919	.968	*.838*	**.893**	.942	**.869**	.899
CRF_0.8	.977	**.938**	**.956**	.978	**.896**	.932	.969	.920	.941
CRF_0.9	.996	.978	.987	.997	.976	.986	.990	.981	.985

Table 4.3: Average results of the proposed models over each split. Italics indicate when the CRF beats the median and bold when the model beats the highest performing model.

Chapter 5

Conclusion and Discussion

5.1 Conclusion

This paper started with the question: Can shallow learning methods learn Named Entity Recognition with little data available? There were two criteria proposed to see whether a shallow learning method could learn with little data available. One was by comparing the results to HIPE 2020 and the other was by seeing at which point a model stops learning.

Although the decision tree model already stops with learning at 10% its overall performance is worse than that of all but two HIPE 2020 teams, one of which being an historical shallow learning method. It also has a clear bias to predicting negatives as it always had a precision of one.

When looking at the HIPE 2020 data set, the CRF beats the median French, German and English performance at 50%, 70% and 30% of the data available. Its top performance is also comparable to the best models of HIPE 2020. The performance of the model keeps improving when increasing the amount of data, only slowing down when reaching a performance of one at roughly 90% of the data. The CRF performs even better when looking at the result of the AJMC data set as it reaches an almost perfect score with only 10% of the data available, with the only exception being English as those results see some improvement. This shows that the amount of data required is not only dependent on the language but also on the domain a model trains on.

So decision trees can quickly reach their top performance whereas the CRF models needed more data to reach it top performance when trained on newspapers, but when trained on commentaries it almost immediately reached top performance.

5.2 Discussion

As already stated in the conclusion the decision tree model showed a clear bias toward negatives. This bias is also present in the data set as there are many more non-named entities then named entities. Although the model might not be a good model for named entities it may prove to be a suitable model for small data sets which are more balanced as it reaches its top performance with very little data available.

CRF on the other hand proved to be suitable for the problem as it can compete with models of HIPE 2020 with fewer data available and as its performance is already close to one it is reasonable to assume that its performance won't increase much if more data was available. But the high results reached in on the HIPE 2020 data set are doubtful as it reaches an higher performance than any of the models used in HIPE 2020 and when looking at historical CRF models they generally did not come close to the results in this paper. In that regard the CRF was trained and tested on an new paper but it reached even higher results with less data available. The high results on the AJMC data make the results on the HIPE 2020 data more likely but the results remain questionable.

5.3 Future research

The decision tree model, the CRF model and the models of HIPE 2020 saw a discrepancy between the results of the different languages which is best seen when looking at the CRF model as the French and English results were higher than the German results with fewer data available. This shows that different languages will need different amounts of data to learn NER. To get more insight into these differences future research could focus on comparing the amounts of data needed to learn NER for different languages. This might provide useful insights into which features of languages determine how much data is needed for NER, which in turn can help the development of NER for new languages. The results of the CRF also showed a difference in data needed when learning on different domains. So besides learning which features of languages impact the data requirement for a model, future research should also focus on which aspects from different domains impact the data requirement.

It should be noted that although the models from HIPE 2020 and the models from this paper had the same training data they did not have the same testing data as the test data set has not been released. It would be reasonable to assume that the test set and the training sets are similar so that testing the models from this paper on the test set from HIPE 2020 would lead to similar results as described in this paper. But, as already mentioned in the previous section, the outstanding results of the CRF model are remarkable. It reaches a performance higher than most of the deep learning models with fewer data available and what makes it extra remarkable is that the IRISA model is also a CRF which doesn't come close to the performance of the model in this paper. This does put some doubt on the validity of the results of this paper, even though high results were reached on a different data set. Therefor it is important for future research to further confirm the results of this paper so that any doubt on the validity of these results can be cast aside.

Besides confirming the validity of the results, future research should also focus on comparing the models from this paper to the models of HIPE 2020 on similar data sets so that a more valid comparison can be made. Along with comparing the models on the same data set, the models should also be compared on different amounts of data to get further insight into the amount of data needed to train each model and to see if different models should be used for different amount of data.

References

CLEF-HIPE. (2022). *Hipe – identifying historical people, places and other entities.* CLEF evaluation lab. Retrieved from https://hipe-eval.github.io/HIPE-2022/

Ehrmann, M., Hamdi, A., Pontes, E. L., Romanello, M., & Doucet, A. (2021). Named entity recognition and classification on historical documents: A survey. *arXiv preprint arXiv:2109.11406*.

Ehrmann, M., Romanello, M., Flückiger, A., & Clematide, S. (2020). Extended overview of clef hipe 2020: named entity processing on historical newspapers. In *Ceur workshop proceedings.*

El Vaigh, C. B., Le Noé-Bienvenu, G., Gravier, G., & Sébillot, P. (2020). Irisa system for entity detection and linking at clef hipe 2020. In *Ceur workshop proceedings.*

Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., … Zhou, Y. (2017). Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*.

Korobo, M. (2017, Jun). *Crfsuite¶.* scikit-learn. Retrieved from https://sklearn-crfsuite.readthedocs.io/en/latest/index.html

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

W3Techs. (2022). *Usage statistics of content languages for websites.* Author. Retrieved from https://w3techs.com/technologies/overview/content_language

Appendix A

Results AJMC

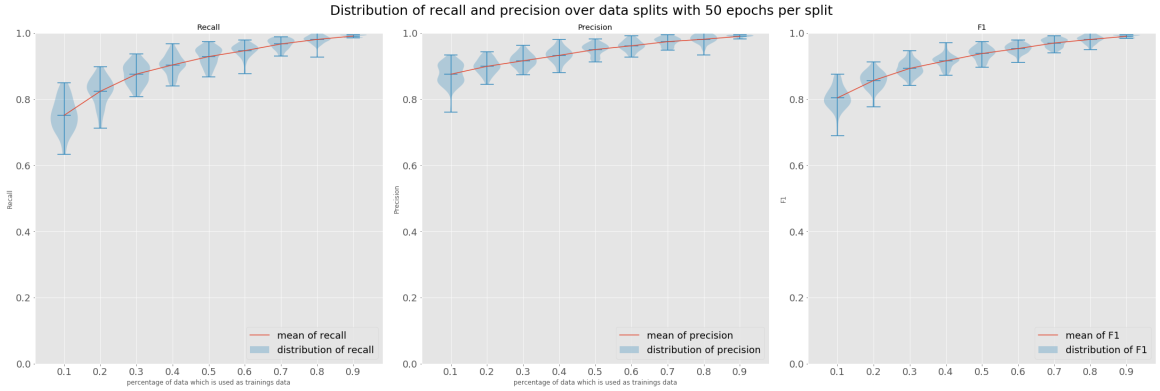

Figure A.1: Results of the CRF model on the English AJMC data

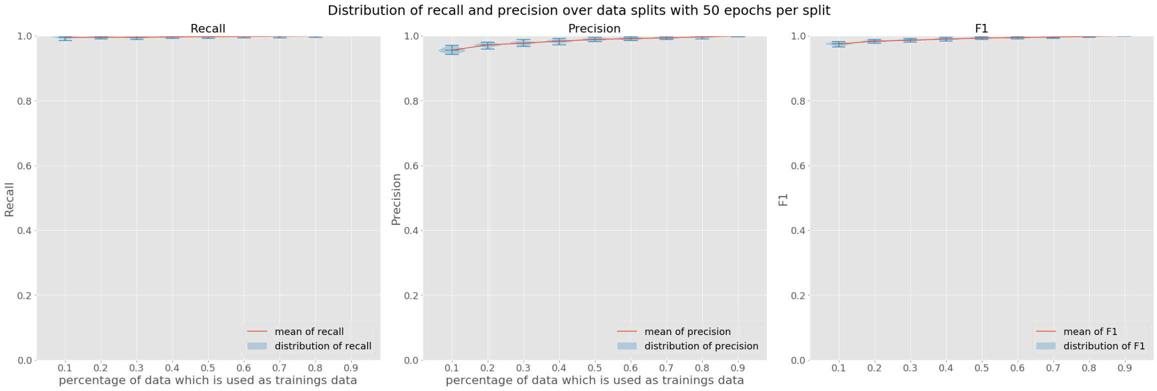

Figure A.2: Results of the CRF model on the French AJMC data

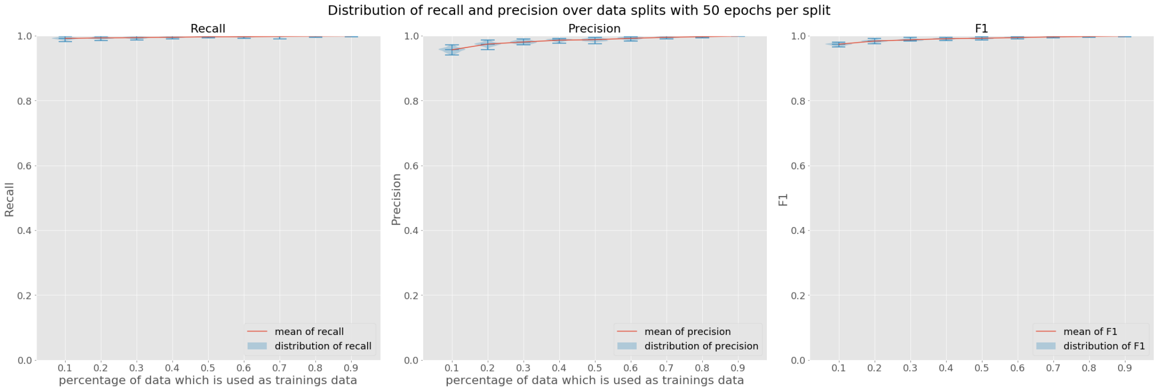

Figure A.3: Results of the CRF model on the German AJMC data