Machine Learning for NLP
Named Entity Recognition task

Bruna Aguiar Guedes (2698211)

Vrije Universiteit Amsterdam

1 Introduction

Named Entity Recognition (NER) is a task from the Natural Language Process (NLP) domain that consists in identifying named entities within text. According to Sang and Meulder [10], "Named entities are phrases that contain the names of persons, organizations and locations". NER is a relevant Information Extraction tool within the field since it can help detecting key elements is unstructured data that gives context (and therefore meaning) to large datasets. The Conll 2003 task on NER, which is the basis of the study, focuses on four types of Named Entities (NE), being them Person, Location, Organization and Miscellaneous entities (NE that do not belong to the previous groups) for an English dataset. For this study we will explore how to find the optimal model with its respective settings, and the construction of features that highlight the model's learning potential.

Here we will first highlight relevant related work developed on the topic on section 2. Then on section 3 a data description including the classes assigned by our own models will be further explained. Next, an overview of the experimental setup is made, including the feature engineering developed to enhance learning of the classifier. On section 5 results of experiments made with the different classifiers are presented, together with two feature ablation experiments, performed in order to understand the effects each feature has on the predictive power of the algorithms being studied. Section 6 is dedicated into an in-depth evaluation of results, and a qualitative approach is used in order to identify misclassification cases as well as situations in which the classifier performs well the task. Finally, in section 7 a discussion of results is exposed, together with the conclusion and a future outlook for the field.

2 Related Work

The Conll 2002 and 2003 shared tasks on NER presented an already existing field of study that remais relevant to the NLP field. While in 2002 advances were made in Dutch and Spanish datasets, in the following year English and German languages were the center of discussion. For the latter, the best model was a combination of four different systems. Years later Qi et al. (2009) [8] trained a neural network with unsupervised learning for the same task showing it was possible to train in an unlabelled corpus.

Other Machine learning techniques have also been proposed, using Conditional Random Fields (CRF), Convolutional Neural Networks (CNN), Long-short Term Memory (LSTM) or even combinations of those, such as Huang et al. (2015) [3] work uniting a LSTM with CRF. More recently, BERT was developed, and among it's classification tasks is NER. BERT is a state-of-the-art model that have a pre-trained corpora of more than 100 languages has boosted the field regarding the language perspective. The BERT Machine Translation approach for cross-lingual NER proposed by Jain et al. (2019) [4] is an example of improvements of annotation-projection approaches for multiple languages including low resource ones. Work on NER in different languages using BERT was developed recently for languages such as Portuguese[11] (mixed with a CRF model), in Arabic [2], as a semi-supervised approach, in Chinese [5] (with enhancements made to the original character-based method) and efforts were made to enhance BERT abilities for NER is low resource languages such as Swedish [6].

Apart from models, the domain factor is also a relevant discussion. While in 2007, Nadeau et al. [7] mentions the lack of studies and negligence regarding the importance of textual genre and domain, this seem to have improved. Studies such as the one from Ritter et al. (2011) [9] focuses specifically on Twitter as a font (being part of the informal genre) and on biomedical data in Spanish [1] (deepening in a domain) are now a reality on the field.

Regarding a NER outlook for future research on the field, Nadeau and al. cite the growing interest in other types of speech that are not only from written text (such as video and oral speech) for NER extraction. Another challenge mentioned is the Machine Translation for NER, which is already evolving, as we could see from Jain et al. research.

3 Data description

The datasets from Conll study being used in this research for the NER task, focuses on four types of NE: Person, Location, Organization and Miscellaneous. The dataset provided also uses a BIO tagging schema (short for beginning, inside, outside), meaning that apart from the aforementioned representation types, instances may be labeled with prefixes. The prefix ’B-’ is used for the first element of a NE. Likewise, a component within the NE from the second element onward receives the prefix ’I-’, and a word which is out of the NE scope, receives solely the label O (in this case no additional information is required). This works under an assumption that NE are non-recursive and non-overlapping.

When specifically referring to the datasets under the scope of this analysis, it can be noticed that the Conll (training and dev) datasets contains the following label scheme: Word (token), part-of-speech (POS) tag, chunk tag and NE tag.

- **Token** Unstructured data tokenized
- **POS** [‘NN’ ’:’ ’NNP’ ’VB’ ’,’ ’IN’ ’DT’ ’.’ ’NNPS’ ’CD’ ’VBD’ ’PRP’ ’JJ’ ’VBP’ ’CC’ ’PRP’ ’VBG’ ’TO’ ’NNS’ ’JJS’ ’WRB’ ’RB’ ’WDT’ ’VBN’ ’POS’ ’RP’ ’VBZ’ ””” ’JJR’ ’WP’ ’O’ ’MD’ ’(’ ’)’ ’LS’ ’SYM’ ’$’ ’FW’ ’RBS’ ’EX’ ’RBR’ ’WP’ ’PDT’ ’UH’]
- **Chunk tag** [‘I-NP’ ’O’ ’I-VP’ ’I-PP’ ’I-SBAR’ ’I-ADVP’ ’B-NP’ ’I-ADJP’ ’I-PRT’ ’B-VP’ ’B-PP’ ’I-LST’ ’B-SBAR’ ’I-CONJP’ ’I-INTJ’ ’B-ADVP’]
- **NE TAG** [‘O’ ’B-LOC’ ’B-PER’ ’I-PER’ ’I-LOC’ ’B-MISC’ ’I-MISC’ ’B-ORG’ ’I-ORG’]

For this study we start by comparing existing systems such as Stanford and Spacy. Regarding the Stanford output for the NER prediction, columns are Word (token), Lower-case token, POS, NE type, and two extra columns containing mostly missing values. Here we don’t have an explicit identification of the B and I tagging, only the O. The BIO tagging can be inferred by the choice of a NE type. Additionally not all NE types match exactly the gold labels scheme.

- **Token** Unstructured data tokenized
- **POS** [‘NN’ ’:’ ’NNP’ ’VB’ ’IN’ ’.’ ’CD’ ’VBD’ ’DT’ ’CC’ ’NNS’ ’TO’ ’RP’ ’PRP’ ’,’ ’RB’ ’MD’ ’JJ’ ’VBN’ ’VBG’ ’PRP’ ’POS’ ’WDT’ ’WP’ ’VBZ’ ’-LRB-’ ’-RRB-’ ’FW’ ’NNPS’ ’LS’ ’JJS’ ’$’ ’VBP’ ’WRB’ ’’̂ ””” ’RBR’ ’JJR’ ’RBS’ ’EX’ ’PDT’ ’UH’ ’SYM’ ’WP’]
- **NE type** LOC, PER, ORG, ‘O’ ’ORGANIZATION’ ’CITY’ ’DATE’ ’NATIONALITY’ ’PERSON’ ’NUMBER”DURATION’ ’STATE_OR_PROVINCE’ ’TIME’ ’LOCATION’ ’ORDINAL’ ’COUNTRY’ ’TITLE’ ’MISC’ ’MONEY’ ’PERCENT’ ’SET’ ’IDEOLOGY’ ’CAUSE_OF_DEATH’ ’RELIGION’ ’CRIMINAL_CHARGE’ ’EMAIL’]

Finally, when evaluating the Spacy output for the NER prediction has the features Word (token), BIO, NE type. Here we see that even though the BIO tag exists, it is not directly linked with the type of NE, nor the types of entities match exactly the gold labels tagging scheme.

- **Token** Unstructured data tokenized
- **BIO tagging** [’B’ ’I’ ’O’]
- **NE type** [’PERSON’ ’ORG’ ’LOCATION’ ’DATE’ ’NORP’ ’CARDINAL’ ’TIME’ ’FAC’ ’ORDINAL’ ’LANGUAGE’ ’LAW’ ’MONEY’ ’EVENT’ ’O’ ’PERCENT’ ’I-PER’ ’QUANTITY’ ’LOC’ ’PRODUCT’ ’WORK_OF_ART’]

4 Experimental setup description

4.1 Preprocessing

Baseline features: After a better understanding of the characteristics of the datasets, and with the goal of comparing systems' performances (w.r.t the gold labels), the next step is to make adjustments in order to align tagging schemes to an unified one.

First, a straightforward decision is made to preserve the gold labels intact, thus realizing the preprocessing steps mainly in the output files (Spacy and Stanford). The only modification to be made to the Conll dataset is labeling the columns, in order to identify columns in a more intuitive manner.

Regarding the Spacy output, all entity types that are not Organization, Person and Location should be grouped into Miscellaneous. Following, a list of conditions is made to group the BIO column with type of entity, which results in the creation of an equivalent column, that is now comparable to the gold labels.

For the Stanford output, similar steps are taken in order to group types of NE into a Miscellaneous group. Since for this output there is no explicit BIO scheme in place for components B and I, an additional rule is created so that if the previous tag is an O, the prefix 'B-' will be added to the type of entity. However, if the previous tag matches the current, it means the word is within the NE chunk, therefore receiving the prefix 'I-'.

The 'equivalence' feature of both datasets as well as the original gold dataset have now, after the preprocessing step the following unique values: ['O' 'B-ORG' 'B-MISC' 'I-MISC' 'B-PER' 'I-PER' 'B-LOC' 'I-LOC' 'I-ORG'].

Extending the system's learning capability: Feature Engineering After an analysis of existing NER classifiers from Spacy and Stanford, as well as our own baseline model, a next step is to now add relevant features that can help the classifier detecting patterns for the NER task. Apart from the given POS-tag and chunk, a feature that detects a capitalized word is extremely relevant for a name Entity in the English language. And the reason is trivial: all name entities, independently of their type will use a capital letter in the beginning (although not all capitals indicate the existence of a name Entity). Thus, a binary feature is created, and a 0 is selected if the token is not capitalized versus a 1 for if it is capitalized.

Additionally, the previous and next tokens are also relevant to the task: for instance, if the previous word is Capitalized and the current one as well, this can indicate that the previous is Inside a NE. Following the same reasoning, a lower case word that follows a Capitalized word may indicate the current token is outside the boundaries of a NE. The input form of the tokens will follow the same logic as the original tokens, since they form the same vocabulary set: each token is converted into a number, and the models will be trained with features represented as one-hot encodings. Later on other options such as the use of word embeddings trained with the Word2Vec language model will also be explored. The advantage of this algorithm is the ability to learn word associations and better understand context on a corpus, however it is still to be proved if its performance is superior than the traditional one-hot encodings. First a SVM model fed only with thos embeddings will be tested, and finally a mixed input will be fed into the same SVM model. This will contain word embeddings regarding the tokens as well as one-hot encoddings for the features POS, and Chunk (which are both original from the initial dataset). For comparison purposes, the same model will be fed with those same features, but this time all in one-hot encoding format so that conclusive findings can be demonstrated.

4.2 Modeling

Baseline model: Logistic Regression As a first approximation to Stanford and Spacy models, we will build our own Logistic Regression classifier that receives only tokens as input features. The choice of the model is based on its simplicity as well as serving the purpose of the classification task.

Since Logistic Regression is initially designed for two-class problems using a binomial probability distribution function to predict the target, a Multinomial Logistic Regression applies to this study, since there are nine possible labels. The classifier takes the list of features as the X and the list of labels as the Y and it fits the model by using a vectorized transformation of the feature (that now acquire the vector format). The cost is calculated using a Cross-Entropy Loss.

Extending to other Machine Learning models As an extension of the baseline model, more complex models are increasingly developed in order to best perform the NER task. A Naive Bayes, although assuming independence between features, can be a good approach for being fast, easy to implement and not requiring a big amount of training data. Differently than Logistic Regression that is a Generative model, Naive Bayes (NB) is discriminative, looking at the target class and then defining the probability of a token happening given that prediction. It's multinomial version is used due to the nature of our multiclass classification problem.

Next, a Support Vector Machines (SVM) is implemented. It is a linear model for classification (as well as regression) that can solve linear and non-linear problems, and in practice has a good performance in many NLP tasks. It's implementation takes an approach of one-versus-rest to predict the target class. It calculates the cost by using a squared-hinge loss, which is a loss used for "maximum-margin" classification, such as the case for SVM. For the model to perform at its best input features are vectorized.

Finally, a BERT model is used, being fine-tuned for two epochs for token classification task. Batch sizes of four are used, as well as a learning rate of 1e-5. For this implementation, only tokens are used as feature, since the model does not require feature engineered additions to have a better predictive power.

4.3 Basic evaluation criteria: Metrics

A basic evaluation will be conducted to compare each system and gold results, using as performance metrics of precision, recall, F1-score and finally performing a confusion matrix to analyse in detail the learning of each system. For the purposes of this whole study, both micro and macro scores were calculated. For the macro scores reported throughout this report the label O was not included in the calculation, since it is relevant to understand how Named Entities are performing. This will be omitted for simplicity of visualization and understanding in the next sections.

5 Results

5.1 Defining a baseline for experiments: Basic system

The performance of Spacy output (seen in Table 1), shows a better prediction of data for the words from outside a NE, probably due to the high number of samples for this class. The system also performs strongly in recognizing the I-PER, with a F1-score of 0.807 and B-LOC, with a 0.780 in the same metric.

When considering precision, the same classes have the best performances, and to this group we add the B-PER with a 0.807 precision. This means not many labels from other classes were mistakenly classified as this tag. On the other hand we have low-performing classes such as B-MISC with solely 0.146 precision. This result is probably partly explained by the fact Miscellaneous tag includes a broader definition, being harder to train. As a final remark, the recall metric for the Spacy prediction have a similar ranking then F1-score.

In what regards the Stanford output, B-PER and I-PER had an impressive performance, with more than 0.92 in all metrics. This indicates the type of NE is very well defined for the classification system. Whilst B-LOC is highly ranked on precision, and it is in the bottom of the rank for recall, indicating the system is focusing much more on not adding false labels then capturing all cases.

	Stanford			Spacy			Baseline		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
B-LOC	0.821	0.138	0.236	0.793	0.768	0.780	0.898	0.746	0.815
B-MISC	0.096	0.783	0.171	0.146	0.636	0.237	0.904	0.594	0.717
B-ORG	0.815	0.864	0.839	0.411	0.334	0.369	0.866	0.512	0.644
B-PER	0.928	0.950	0.939	0.807	0.675	0.735	0.871	0.509	0.642
I-LOC	0.873	0.241	0.378	0.462	0.568	0.510	0.783	0.479	0.594
I-MISC	0.072	0.523	0.126	0.056	0.382	0.098	0.868	0.382	0.530
I-ORG	0.901	0.603	0.722	0.385	0.635	0.480	0.851	0.313	0.458
I-PER	0.974	0.919	0.946	0.819	0.794	0.807	0.794	0.180	0.293
O	0.989	0.830	0.902	0.967	0.854	0.907	0.919	0.999	0.958
Macro scores	0.685	0.628	0.545	0.485	0.599	0.502	0.854	0.464	0.587

Table 1: Evaluation metrics Precision, Recall and F1 score when comparing gold labels and the Stanford prediction, the Spacy prediction, or the Baseline system

Finally, when evaluating our own basic system that will serve as baseline for experiments, we see that if only tokens are used as input feature and a Logistic Regression is applied, F1-scores as low as 0.293 can be found (for I-PER class). For all cases of our system, the B- category from the BIO tagging scheme was better classified than the classes with prefix I-. The macro F1-score of the Baseline reached 0.587, performing better than the 0.545 from Stanford classification and the 0.502 performance from Spacy. If however the macro score include all classes, this metric achieves 0.628 for the baseline model.

Confusion Matrices: As a next step towards understanding the strengths and opportunities of each system, a Confusion Matrix was calculated for each model to provide a better overview of which categories within the models had a higher or lower performance.

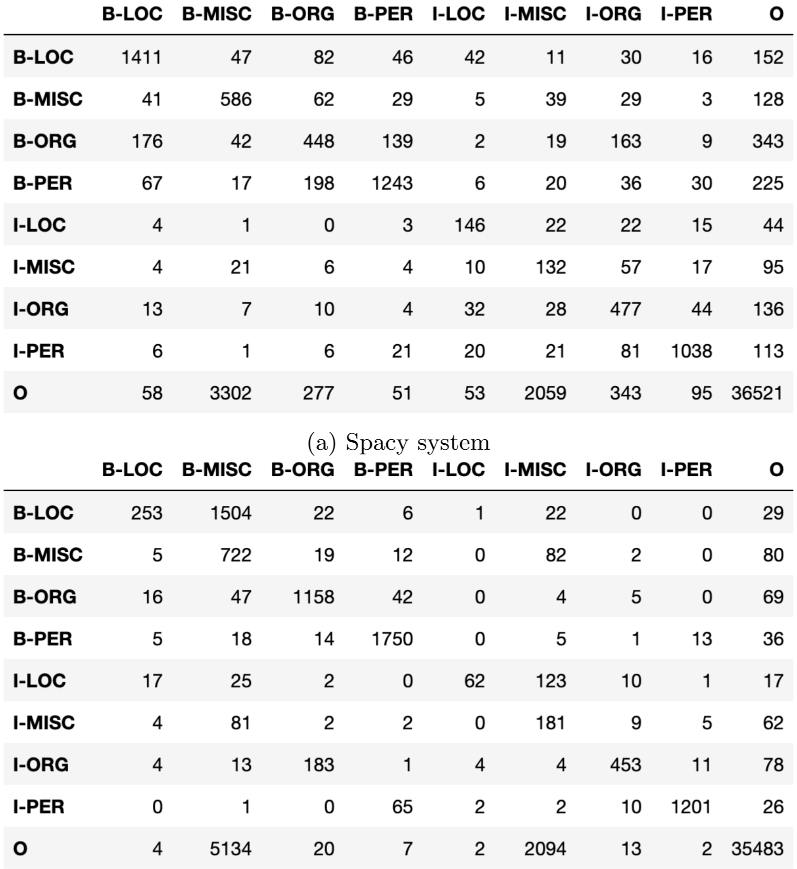
The results for Spacy shown in Figure 1a points that the O class - which is made of all non-NE - is more easily misinterpreted with among all classes, partly because it lacks a strict definition. It is also unsurprising how this category was mainly misclassified as Miscellaneous - which also counts with a broader definition. The other classes performed considerably better, with more cases of mislabelling between B-ORG and B-PER and between B-ORG and I-ORG.

Stanford output results are shown in Figure 1b. Although the same misclassification of O applies to this output, in general the true positives are more easily recognizable in the confusion matrix. Errors in labels are 0 in several occasions, especially for cases with I- prefix.

5.2 Extending to new features and other Machine Learning models

After this first approximation to developing a NER system, two steps are taken: i) the inclusion of features that might help the classifier to learn better, and ii) the use of benchmarks, which include A Naive Bayes approach and a Support Vector Machines implementation, both without fine-tuning. When using tokens, POS tags, chunks, previous word, following word, and identifying capitalized words, there is not only an expectation of improvement in the models, but also a better chance to view the different model behaviors. For this reason, at this step of experimentation we assume all features have a relevant role in the learning - this assumption will be tested next.

First, to the baseline model, we now input the complete set of features extracted from the tokens. The Logistic Regression with the new input changed abruptly from the previously seen 0.587 f1-score to a 0.928, and it's poorest performance class-wise was a 0.90 now for I-ORG label. The Naive Bayes is the worst performer among the systems developed, with a macro performance of 0.914, but a recall of 0.529, indicating the classifier is missing many true cases. For I-LOC, this model had only 0.015 score, which translates into two correct labels from the validation set. The same model did, however achieve a 0.906 f1-score for I-PER. Those results are not unexpected. Since the input contains highly correlated features and Naive Bayes assumes independence between features, this causes that



	B-LOC	B-MISC	B-ORG	B-PER	I-LOC	I-MISC	I-ORG	I-PER	O
B-LOC	1411	47	82	46	42	11	30	16	152
B-MISC	41	586	62	29	5	39	29	3	128
B-ORG	176	42	448	139	2	19	163	9	343
B-PER	67	17	198	1243	6	20	36	30	225
I-LOC	4	1	0	3	146	22	22	15	44
I-MISC	4	21	6	4	10	132	57	17	95
I-ORG	13	7	10	4	32	28	477	44	136
I-PER	6	1	6	21	20	21	81	1038	113
O	58	3302	277	51	53	2059	343	95	36521

(a) Spacy system

	B-LOC	B-MISC	B-ORG	B-PER	I-LOC	I-MISC	I-ORG	I-PER	O
B-LOC	253	1504	22	6	1	22	0	0	29
B-MISC	5	722	19	12	0	82	2	0	80
B-ORG	16	47	1158	42	0	4	5	0	69
B-PER	5	18	14	1750	0	5	1	13	36
I-LOC	17	25	2	0	62	123	10	1	17
I-MISC	4	81	2	2	0	181	9	5	62
I-ORG	4	13	183	1	4	4	453	11	78
I-PER	0	1	0	65	2	2	10	1201	26
O	4	5134	20	7	2	2094	13	2	35483

(b) Stanford system

Fig. 1: Results of confusion matrix when comparing the gold labels (rows) with the specific system (columns)

similar information is computed multiple times in the model, augmenting its real importance and thus degrading the model.

Furthermore, results for the SVM model were also extracted. An overall precision of 0.995 was achieved, as well as a recall of 0.989. In all classes for the three metrics results were above 0.979 for the validation set, indicating a quite successful learning for this algorithm with the selected input features.

Finally, a extension to BERT model takes place, and for this pre-trained model, no additional features are required. After two epochs finetuning the model with specifications described in the Modelling section, we see that the final loss of the validation set is of 0.05 and F-1 macro score is 94.15. This performance is higher than our Logistic Regression model with all input features by 1.35%, however it is lower then our complete SVM model by 5.05%.

5.3 Feature optimization

Apart from the traditional features, represented though the use of one-hot encodings, a SVM model trained with word embeddings from Word2Vec language model was used. While precision had a performance of 0.775 and recall of 0.669, the total F1-score was of 0.713. Class 'I-ORG' was the poorest label prediction, as it was the case of SVM with one-hot encoddings. On the other end, we have B-LOC as the highest prediction performance, with a 0.856.

The mixed models was a mid-term that includes word embeddings for tokens as well as one-hot encodings for the original features POS and Chunk (due to computational power it was not possible to include all variables developed). When compared to the word embeddings only, the mixed model is an upgrade. Also using an SVM model, this input format has achieved a 0.779 f1 macro score, and its worst perfoming label (which remains being I-ORG) now increases to 0.663. The class with best predictions has now shifted to B-PER, with a 0.880 performance. For comparison purposes, if we use the same SVM model and input the same features Token, POS and Chunk, with the only difference being they are now all one-hot encodings, F1-score goes to 0.887, indicating a far superior predictive power from the traditional encoding.

Ablation analysis Up to this point the previous algorithms were tested under the assumption they contained insightful and relevant features. Although this step was interesting to reveal contrast between the models, a feature ablation analysis is made on top of the traditional systems (i.e the ones fed with one-hot encoddings), in order to inspect the real relevance of each individual feature and how they relate amongst themselves. In Table 2, we can evaluate the individual impact of each feature, when input together with tokens only for Logistic Regression system as well as for the SVM. In both cases, the most insightful features for the learners is the Previous token, that improved by 19.4% the performance of the Logistic Regression model and 10.2% for the SVM. Zooming into the Logistic Regression, the POS tagging and the Following feature were also highly relevant, together with token both cases presented a 76.5% and 73.0% performance respectively. Chunk and Capital features when individually added to token input were not that informative, and the Capital added solely 0.7% on the overall performance of the Logistic Regression model.

Regarding the SVM, is it interesting to notice that by adding only the single feature Previous, performance is quite similar from the complete model (that achieved a 99.2% F1-score with 6 features versus the 95.7% with two features, token being one of them). When adding Following feature to the tokens, there is a similar effect (94.8%). The features POS and Capital were here the least relevant, with an improvement of 0.7% and 0.1% respectively.

In Table 3 we can observe the effect of applying a step down strategy for feature selection applied to the Logistic Regression model. Its performance and the percentage of change in its macro F1-scores are calculated and here we can observe the interaction effect of the features. The criteria for removal was based on the ranking, removing from less informative features to most informative ones based on the individual analysis. When removing Capital and Chunk the performance remains piratically unchanged, indicating it is a better trade-off to remove them and having a simpler model. Removing

Following and POS already have a more important effect on prediction since f1 drops 5.8% and 8.5% respectively. Finally, as previously seen, by removing Previous and leaving only the token is the biggest gap, affecting deeply the performance.

The same procedure is put into place for the SVM model, and based on its ranking features were removed one by one. The model receiving only the token itself is already quite informative, so gaps when removing features are not as sizeable. Removing Capital, POS and Chunk is a worth trade-off since the model loses only 0.3% of its explainability. By also removing Following an extra 3.3% of performance is lost. Finally by leaving the token only as input hurts the algorithm the most percentage-wise.

	LR	**Rank LR**	**SVM**	**Rank SVM**
Token + Previous	0.781	1	0.957	1
Token + POS	0.755	2	0.862	4
Token + Following	0.730	3	0.948	2
Token + Chunk	0.649	4	0.883	3
Token + Capital	0.594	5	0.856	5
Token	0.587	6	0.855	6

Table 2: Macro f-1 score for individual feature importance analysis for SVM and Logistic Regression. Here Rank indicates the Ranking from more important features to the system to least important.

	Features Removed	LR	% ΔLR
COMPLETE	None	0.928	-
RM1	Capital	0.927	-0.1%
RM2	Capital, Chunk	0.925	-0.2%
RM3	Capital, Chunk, Following	0.867	-5.8%
RM4	Capital, Chunk, Following, POS	0.781	-8.5%
TOKEN ONLY	Capital, Chunk, Following, POS, Previous	0.587	-19.5%

Table 3: Macro f-1 score for ablation analysis for Logistic Regression. Here RM indicated the number of features removed.

	Features Removed	SVM	% ΔSVM
COMPLETE	None	0.992	-
RM1	Capital	0.992	0.0%
RM2	Capital, POS	0.991	-0.2%
RM3	Capital,POS, Chunk	0.990	-0.1%
RM4	Capital,POS, Chunk, Following	0.957	-3.3%
TOKEN ONLY	Capital,Pos, Chunk, Following, Previous	0.855	-10.2%

Table 4: Macro f-1 score for ablation analysis for SVM. Here RM indicated the number of features removed.

6 Error Analysis

An error analysis is conducted next in order to better understand how to improve and what are the strengths of the best Logistic Regression model (fed with six input features). A confusion matrix can be see in Table 5. Here we observe that most False Positives are detected with the misclassification of classes O and B-LOC (in order). The Type I error occurred 483 times for the first and 91 times for the latter. I-PER has also suffered from 57 False Positives. In the three classes mentioned the error was spread throughout all classes of the true labels. It is also noticeable the good performance of the model regarding all other classes with prefix 'I-'. Solely 4 False Positives happened for both I-LOC and I-MISC and 11 for I-ORG.

Type II errors were also present in the predictions. For the gold label B-MISC 130 instances were mislabeled, and for 'I-ORG' the missed cases were 127. If we disregard the class O since it is not part of a Name Entity, the mislabeling drops significantly. In this scenario both previously mentioned classes still have the highest literal numbers for mislabeled data, however they represent 42 instances for I-ORG and 38 for B-MISC.

After this first examination of results, a qualitative analysis takes place, and the misclassified instances are inspected in order to understand the possible reasons of disorientation.

False Positives (Type I error):

– Prediction of class B-LOC:
  - B-ORG gold label:
    For cases in which Organizations have the same name of a Location the predictor mislabeled B-ORGs per B-LOCs. For instance in the context of sports Milan and Torino (which are both cities in Italy), were not properly contextualized as being Soccer teams, and therefore B-ORG. Another similar example is the word Philadelphia, that appears 12 times in the dataset, 5 of them with gold labels B-ORG and 7 with gold labels B-LOC. This indicates a reason why the model might have leaned to the wrong labeling on the specific scenario in which Philadelphia refers to an Organization.

    (i.) *...Gianluigi Lentini, transferred to Milan in 1992...*
    (ii.) *...Los Angeles 7, Philadelphia 6.*

  - I-ORG gold label: In cases for which a name of a country appears and the previous word is a preposition, the classifier could not identify (since it did not have the feature) that two previous words before a name of an Organization (in all cases from the dataset political parties) have started. Therefore it has labeled the token with the country name as B-LOC instead of I-ORG

    (i.) *...the rally for the Return of Refugees and Democracy in Rwanda (RDR)...*
    (ii.) *...the New York-based group Human Rights in China...*

– Prediction of class O:
  - B-MISC gold label: In cases of hyphenated adjectives, it seems like the classifier is not able to predict if a word is part of a NE. Several scenarios occurred in which the adjectives are considered to be outside such as in the following word constructions: *'Ohio-based', 'Bergamo-based', 'mainly-Moslem', 'Kurdish-controlled'.*
  - B-PER gold label: Perhaps due to very low repetition in dataset, the classifier mislabeled 76 instances of B-PER as outside of the NE. Apart from that it seems harder to find a pattern for instance on the Previous or Following features, being them from a big range of parts of speech (from punctuation, to verbs, substantives, prepositions, etc). Thus it is harder to build an intuition from an unknown word with no clear pattern on other features either.

False Negatives (Type II error):

- True class B-MISC: A general misclasification of the gold label B-MISC is related to origin of people. Those should be classified as B-MISC, however there doesn't seem to be a clear criteria from the classifier to predict. While the token 'English' was considered as B-ORG, 'Brazilian', 'Spanish', 'Austrians' and 'Pakistanis' were not even considered as a NE, 'Welsh' were considered as B-PER in one example and O in another.

(i.) *Rugby Union: English, Scottish and Welsh results.* Welsh predicted as 'O'
(ii.) *Although winger Pieter Hendriks appeared toknock on Joubert's reverse pass, Welsh referee allowed...* Welsh predicted as 'B-PER'

		Prediction from LR								
		B-LOC	**B-MISC**	**B-ORG**	**B-PER**	**I-LOC**	**I-MISC**	**I-ORG**	**I-PER**	**O**
Gold	**B-LOC**	1756	2	7	8	2	0	2	4	56
	B-MISC	12	792	8	11	0	2	0	5	92
	B-ORG	22	9	1221	10	0	0	1	4	74
	B-PER	8	1	5	1748	0	0	0	4	76
	I-LOC	6	0	0	0	222	0	4	10	15
	I-MISC	2	4	0	3	0	274	3	6	54
	I-ORG	23	0	4	0	2	1	624	12	85
	I-PER	3	0	1	5	0	1	0	1266	31
	O	15	2	7	47	0	0	1	12	42675

Table 5: Confusion matrix for Logistic Regression model with 6 input features

7 Discussion and Conclusion

In this study we explored different models to build a Name Entity Recognition System. We have built a baseline from a Logistic Regression model that received only tokens as input feature and evolved to more complex models, as well as more complex features. For the first we have trained a Naive Bayes, evolved to a SVM, and compared it to a BERT. For the latter, features that capture important aspects of the token were chosen, as well as different formats were explored (by using one-hot econdings and word embeddings).

The best performer considering all aforementioned models, features and input formats was the SVM with six features input as one-hot vectors. This system captures most of cases from our dataset, and achieved a 0.992 f1 macro score. It is important however, to evaluate the trade-off between number of features (i.e complexity) and performance. For this specific case, removing four less relevant features impact in only 3.3% of predictive power, while making the model more straightforward.

When testing state-of-the-art models such as BERT, performance after only two epochs was already quite impressive (considering there is no need to input any additional features). However this study proves that BERT is not always the optimal model, depending, among other things, on the setup (e.g: learning rate, batch size, optimizer, epochs).

Another important finding is the power that relevant features in the correct format can have to the learning of a system. When the models were first input with token only (for instance the Logistic Regression baseline), performance was of only 0.587. However, by improving the input, the same model achieved 0.928. Those results were also seen for SVM in a similar manner. The only exception to this was the Naive Bayes model, which can be explained by it's assumption of independence that is not fulfilled when we add correlated features to the input.

Finally, the feature format has also affected models learning. A SVM with only tokens as a one-hot encoding had a performance of 0.855, while by changing it to word embeddings (in this case by using the Word2Vec language model) results dropped significantly to 0.713.

Much can still be explored as a next step to this research. It seems relevant to compare in the future how this models (and more especially features) would react in different domain areas such as ancient literature, or technical texts.

References

1. Akhtyamova, L.: Named entity recognition in spanish biomedical literature: Short review and bert model. In: 2020 26th Conference of Open Innovations Association (FRUCT). pp. 1–7. IEEE (2020)
2. Helwe, C., Dib, G., Shamas, M., Elbassuoni, S.: A semi-supervised bert approach for arabic named entity recognition. In: Proceedings of the Fifth Arabic Natural Language Processing Workshop. pp. 49–57 (2020)
3. Huang, Z., Xu, W., Yu, K.: Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
4. Jain, A., Paranjape, B., Lipton, Z.C.: Entity projection via machine translation for cross-lingual ner. arXiv preprint arXiv:1909.05356 (2019)
5. Jia, C., Shi, Y., Yang, Q., Zhang, Y.: Entity enhanced bert pre-training for chinese ner. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 6384–6396 (2020)
6. Malmsten, M., Börjeson, L., Haffenden, C.: Playing with words at the national library of sweden–making a swedish bert. arXiv preprint arXiv:2007.01658 (2020)
7. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes **30**(1), 3–26 (2007)
8. Qi, Y., Collobert, R., Kuksa, P., Kavukcuoglu, K., Weston, J.: Combining labeled and unlabeled data with word-class distribution learning. In: Proceedings of the 18th ACM conference on Information and knowledge management. pp. 1737–1740 (2009)
9. Ritter, A., Clark, S., Etzioni, O., et al.: Named entity recognition in tweets: an experimental study. In: Proceedings of the 2011 conference on empirical methods in natural language processing. pp. 1524–1534 (2011)
10. Sang, E.F., De Meulder, F.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. arXiv preprint cs/0306050 (2003)
11. Souza, F., Nogueira, R., Lotufo, R.: Portuguese named entity recognition using bert-crf. arXiv preprint arXiv:1909.10649 (2019)