**Dataset - NERC test data**

The NERC test data consisted of only 12 sentences, which corresponds to 164 labeled instances. The majority class is O with 71.34% of the dataset, followed by I-per with 7.32%, then B-per with 6.1%, B-org with 4.27%, I-art with 3.66% and I-org with 3.05%. The rest of the classes all make out below 2% of the dataset each. There are 9 classes in total.

**Preprocessing and Feature Selection**

For the Named Entity Recognition and Classification (NERC) we used a dataset from Kaggle.

The preprocessing steps were kept simple, with splitting the sentences into separate words. Within the feature extraction the words themselves were transformed into lower case, however, the information weather a word was initially capitalized was preserved as this might give important insights for recognizing entities.

Standard features for this task were used, namely the word in lower case, the word suffix of three and 2 characters, whether the word is capitalized, is in title script, or is a digit, the Part of Speech (PoS) tag of the word and the first 2 characters of the PoS tag. For the PoS tagging, the .tag\_ feature from spacy was used. Additionally, to the features of the current word, the lowercase form, if the word is capitalized or in title case, the PoS tag and the first two characters of the PoS tag were added for the word preceding and succeeding the current word. If no one preceded the current one, the current one, the BOS label was added to the features instead to signal the beginning of the sentence. Similarly, if there was no following word in a sentence, the EOS (end of sentence) tag was added to the features.

A test set (20% of the data) was split from the data, the remaining data was split into a validation set (20% of the remaining data) and a training set. The test set was only used to measure the performance of the final model, the validation set was used for hyperparameter tuning and selecting the best model. In the end the best model was also evaluated on the test data that was provided for the assignment.

**Dataset**

The test set included 9,578 sentences which yielded 209,994 tokens/labels, the biggest class of the labels was O with around 84.74%. The next 5 classis in descending order were B-geo with 3.59%, B-tim with 1.95%, B-org with 1.91%, I-org with 1.53% and I-per with 1.64%.

In terms of size, the validation set included 7,662 sentences, corresponding to 166,841 word tokens. Again, the majority class was O with 84.62%, succeeded by B-geo with 3.53%, B-tim with 1.97%, B-org with 1.95%, B-per with 1.61% and B-gpe with 1.6%.

As the biggest set, the training data includes 30,650 sentences, which translates to 670,156 BIO labels. Unsurprisingly O is the biggest class with 84.62% of the tokens. Next are B-geo including 3.62% of the labels, B-tim with 1.93%, B-org with 1.92%, I-per with 1.66% and B-per with 1.64%.

All sets are unbalanced with O as the majority class and B-geo including a bigger sample than the other B and I labels. We decided against artificially balancing the classes. While this might result in better results for the dataset that is trained on, this artificial balance would not correspond to any real world scenario where the majority of words would indeed not be named entities (corresponding therefore to an O in terms of BIO tags). The distributions within the training, test and validation sets are all similar, making accurate performance metrics possible.

**Method and Model**

For the actual NERC task, we trained a conditional random field (CRF) model with the modules from sklearn. Conditionally random field for IOB sequences maps sequence dependencies between tokens. Therefore, labels are not just assigned based on the current token, but based on sequential paths.

There was extensive hyperparameter tuning for the CRF model using a grid search method. While the feature engineering stayed constant, the search space fitted models with the optimization algorithms lbfgs (limited memory stochastic gradient descent), l2sgd (stochastic gradient descent with L2 regularization), ap (Averaged Perceptron), pa (Passive Aggressive). For both the L1 and L2 regularization parameters the values 0.2, 0.1, and 0.01 were tested. The models were trained with and without all\_possible\_transitions activated. If it is activated, transition features are generated that do not occur in the training set. All possible combinations of the named values were computed and compared by the weighted average of their f1-score (calculated for the validation set). The impossible combinations were skipped (for example l2sgd does not support L1 regularization). One hyperparameter that was not tunes is the maximum number of iterations, which was fixed and set to 100. Since the weighted average f1 score can be misleading due to the high class imbalance, we also optimized with regards to the macro average f1 score.

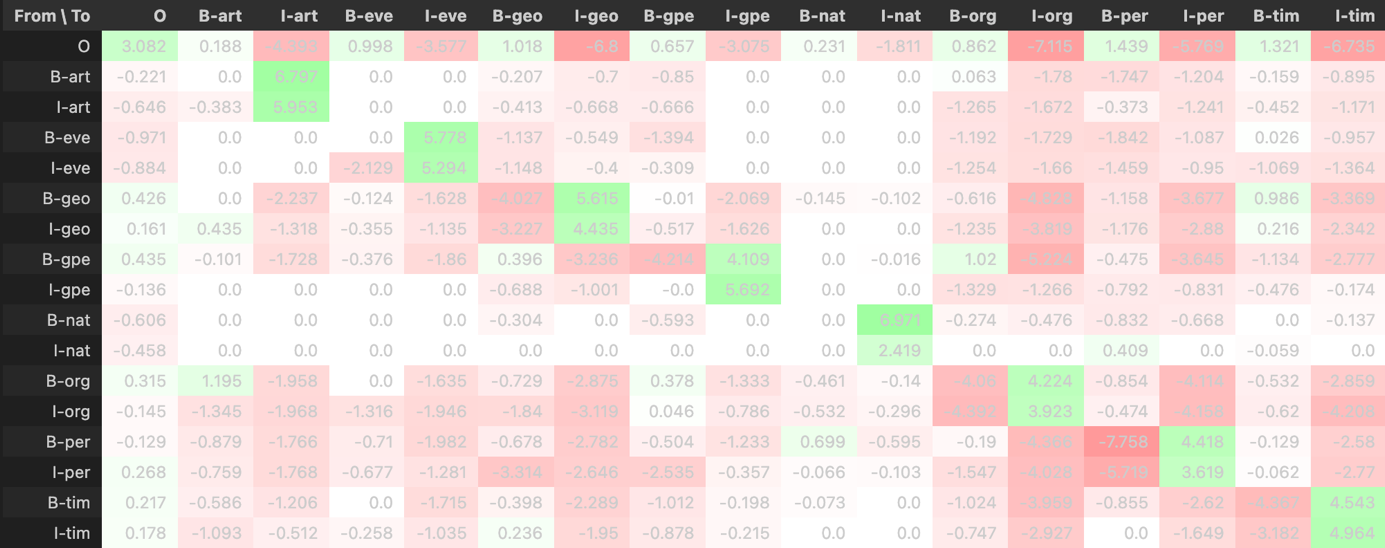
One limitation of the method is that more parameters were tested for the algorithms working with gradient descent (lbfgs and l2sgd), giving them more possibilities to achieve good results.

**Analysis**

The best model was achieved with the lbfg algorithm, an L1 regularization parameter of 0.2 and an L2 regularization parameter of 0.1. all\_possible\_transitions was set to the Boolean value True, so the feature was activated. For both the macro and the weighted average f1 score this was the highest performing setup.

This best model was evaluated first on the validation set and then on the test set. The values for the validation and the test set show a lot of similarity with some values even being slightly better for the test set. This is an indication that the model did not overfit to the training set. The following values are all from the evaluation on the test set. With 0.97, the model achieved a high overall accuracy. The macro average for the precision is 0.73, for the recall is 0.62 and for the f1-score is 0.66. With regards to the weighted average, the precision, recall and f1-score values are all 0.97. O reached the overall the highest precision, recall and f1-scores with 0.99 each, which was to be expected seeing as O is the majority class.

Outliers with regards to especially low performance metrics were the classes I-art (precision: 0.29, recall: 0.1, f1-score: 0.15) as well as B-art (precision: 0.24, recall: 0.07, f1-score: 0.11). This is most likely doe to the low support of the art class in general, especially with regards to I-art, which only has 210 training instances in total and only 59 instances in the test set. For the same reason the B-nat and especially I-nat classes do not perform well, with B-nat having a precision of 0.53, a recall of 0.40 and an f1-score of 0.47 with a support of 72 in the test and 135 in the training set. I-nat performs even worse with a precision of 1.0, a recall of 0.4 and an f1-score of 0.57. It has a support of only 24 in the training data and 5 in the test data. While the precision score of 1 looks perfect at a first glance, this probably only means that none of the 5 instances were false positives. The metrics for this minority class are most likely not better than random guessing would be.



When visualizing the weight for the CRF model it can be seen that the model learned high positive correlations from B-eve to I-eve, from B-geo to I-geo, B-grp to I-grp, from B-art to I-art, from B-nat to I-nat, from B.org to I-org, from B-per to I-per and from B-tim to I-tim, which are aspects which we wanted the model to realize. Furthermore, it can be seen that especially negative connections are formed from O to I-geo, I-org, I-per and I-time, which makes sense since O would be followed by a B, which only then would be followed by an I label. Moreover, B-grp is not often followed by B-grp, indicating that groups often have a one-word name, from B-grp to I-org, signaling that group entities are not often followed directly by organization entities. The negative weights from B-per and I-per to B-per might indicate that in the training data, persons were often only having a single word name (as supposed to a full name consisting of first and last name directly following each other).

Lastly, the model was tested on the test data that was specifically provided for NERC. Here the model had a lower accuracy of 0.84. The weighted averages are 0.82 for the precision, 0.84 for the recall and again 0.82 for the f1-score. These values are probably highly skewed due to the metrics of the majority class O, which has a precision of 0.91, a recall of 0.96 and an f1-score of 0.93. Macro averages show a more balanced view on the data and are worse that the weighted averages with precision and f1-score only lying at 0.43 and recall being slightly better with 0.45. However, this data might be impacted by the three classes that do appear in the test set and therefore return 0 for precision, recall and f1-score. Taking this into account and ignoring the values where there is a zero-division, the weighted averages are precision: 0.87, recall: 0.84. The macro averages improve even more with precision: 0.52, recall: 0.6. The f1 score does not change, since these empty classes are still (wrongly) predicted by the system, even though they don’t exist in the test set. An outstandingly good performance can be seen got I-geo, where precision, recall and f1-scores are all 1, indication perfect performance. However, since the support for this class in only 2, this could also be based on luck. Low performance can especially be observed in classes that already performed badly on the prior test set, namely B-art and I-art. All of the values for art are 0, indicating that no instance was classified correctly. The second biggest class in the test set, I-per performed relatively well with a precision of 1, a recall of 0.58 and an f1-score of 0.74. The equivalent B-per has well and more balanced results with a precision of 0.73, a recall of 0.8 and a f1-score of 0.76. In general, the small size of the test set makes the performance evaluation less significant.