Evaluation of the Conditional Random Field POS-Tagger

Simon Bross
University of Potsdam
bross@uni-potsdam.de

Abstract

The following evaluation report examines the Conditional Random Field model - a discriminative model for segmenting and labeling data - adopted for the task of part-of-speech tagging. To get a general idea of the model's performance, 4 datasets of different size and language (German and English) are used to train and evaluate the model employing different metrics and procedures, including baselines, hyperparameter optimization, k-fold cross validation, and a linguistic error analysis that aims to uncover what the metrics hide under the guise of numerical values. In the single train-test split, the non-optimized CRF model (good baseline) performed best on the second largest dataset (EWT, English) with an F1 value of 0.87. The hyperparamater optimization gave only a slight and negligible improvement in the single train-test split procedure (F1 of 0.88) and k-fold cross validation (maximum F1 of 0.83), thus requiring a more thoroughly conducted optimization. 10-fold cross validation on the small English and German datasets showed that the model performs (slightly) better when the data is split into disjoint folds that contain randomly distributed data, but also that the PUD ENG dataset is unbalanced due to its varying performance between the folds.

1 Introduction

Although the evaluation of an NLP model may be a costly - in terms of both time and money - and complex task, the resulting data and the experience gained from it, if carried out with consideration, are indispensable for the assessment, comparison, and understanding of the model in question. It is not only individuals who are interested in a concise and transparent model evaluation, but also funding bodies, the research community, and industrial companies, all of which may have different needs and expectations.

However, common ground expectations focus on decision-making processes such as the selection of a system from a set of competing ones. Evaluation reports of NLP models thus provide findings as well as objective and subjective evidence that allow individuals or institutions to determine whether and to what extent a system's qualities and limitations can satisfy their requirements.

In addition to providing evidence for decision-making, Margaret King points to a desired adaptability of a system: "Evaluation is aimed at finding out not only what the system currently does but also how easily it can be modified" (King, 1996, p. 74). This not only applies to newly built systems but also particularly to established ones, where evaluation serves as a means of capturing and documenting progress, thus being indicative of whether adjustments might be necessary.

For the research community, evaluation serves as an indicator of the status quo during every evaluation/research stage and how further research should proceed rather than as a mere (show-off) performance measure, given that the underlying motivation for this evaluation consists in producing valuable and relevant knowledge and "convinc[ing] the community that your ideas are worthwhile, that they work and how" (Cohen and Howe, 1988, p. 35). Therefore, evaluation is a crucial component of the research process, allowing the researchers to determine the effectiveness and impact of their work as well as to guide future research aiming to refine methods and produce more robust and meaningful findings on a system.

2 Task

The NLP task evaluated in this report is **part-of-speech tagging (POS tagging)**, an essential and historically used sequence labeling task in natural language processing. Parts of speech, such as

nouns, verbs, and adjectives, provide information to identify the grammatical category of each word in a sentence and help to disambiguate words with multiple meanings by providing context. Depending on the implementation, the input for a POS tagger consists either of a sequence of words (i.e. a tokenized sentence) or a sequence of sentences (i.e. several tokenized sentences), where each token from a sentence is mapped to a tagset-specific speech tag, thus outputting either a sequence of labels (tags) that each corresponds to a token from the input sequence (cf. 1 below), or a sequence of labeled sequences.

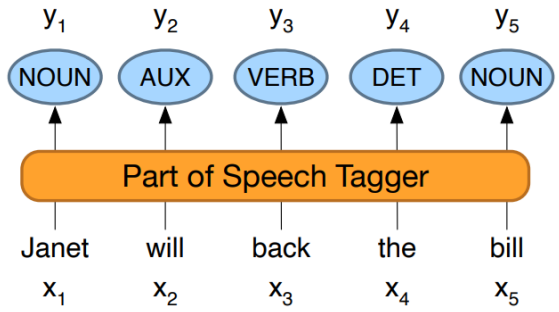

Figure 1: Input (X_n) and output (y_n) of a POS tagger exemplified by the sentence "Janet will back the bill" (Jurafsky and Martin, 2020, p. 152)

POS tagging is used in a wide variety of NLP systems whenever grammatical information is beneficial or necessary for processing language and is often employed as a corpus preprocessing technique. It can be further used for lemmatization, which is a system that reduces inflected forms to their underlying lexemes/lemmas, thus disambiguating homographs and homonyms (Srinivasa-Desikan, 2018). Further relevant applications are, amongst others, **Named Entity Recognition (NER)**, which uses POS tags to identify proper nouns, such as names of people and places, **Sentiment Analysis**, where POS tagging helps to assess the polarity of a word, and **Question Answering**, using POS tags to analyze both the presented question and the answer to be produced (ibid.). Thus, POS tagging is considered an outstanding and ubiquitous groundwork tool in NLP (Chiche and Yitagesu, 2022).

3 Model

The **Conditional Random Field (CRF)**, "a framework for building probabilistic models to segment and label sequence data" (Lafferty et al., 2001, p. 282), is used to train the POS tagging models. It was invented by Lafferty et al. in 2001 motivated by the general necessity "to segment and label sequences [...] in many different problems in several scientific fields" (ibid.).

The model was proposed as an alternative to generative models such as **Hidden Markov Models (HMMs)** as well as stochastic grammars, attempting to overcome the limitations of the previous generation of sequence labeling models. The reason I chose this model is that I have never delved deeply into a classical sequence labeling algorithm before other than HMMs. The motivation is therefore rather epistemic, striving to deepen my understanding and expertise in classical sequence labeling algorithms beyond HMMs. Even though modern state-of-the-art sequence labelers based on neural networks are gradually supplanting the classical algorithms, it is still important, and even more so in light of the downsides of neural networks, to uderstand and experiment with classical models.

The NLTK's (Bird et al., 2009) CRF module (API Reference, Source code) is used to build the models, which is a wrapper around the sklearn (Pedregosa et al., 2011) CRFSuite Tagger (API Reference, Source code). In essence, the CRF algorithm works by "comput[ing] log-linear functions over a set of relevant features, and these local features are aggregated and normalized to produce a global probability for the whole sequence" (Jurafsky and Martin, 2020, p. 163). These features are, for instance, word prefixes and suffixes of varying lengths indicative of certain parts of speech, such as the suffix '-ing' is indicative of that the respective word may be a verb in present participle form. By using feature functions that retrieve morphological information about a word in the input sequence, the CRF algorithm has a clear advantage over HMM models that struggle to accurately predict unknown words, proper names, and acronyms.

4 Data

The model training and evaluation is performed using four different datasets from the Universal Dependencies, which is "a framework for consistent annotation of grammar [...] across different human languages" (Universal Dependencies, 2022e), that aims at "facilitating multilingual parser development, cross-lingual learning, and parsing research from a language typology perspective" (Zeman et al., 2022).

4.1 German Part of the Parallel Universal Dependencies (PUD)

This treebank consists of 21.000 tokens and 21.331 words in 1.000 German sentences, released under the CC BY-SA 3.0 license and created for the CoNLL 2017 shared task on Multilingual Parsing from Raw Text to Universal Dependencies (cf. Universal Dependencies, 2022e). The PUD treebank sentences were randomly selected from the news domain and Wikipedia. The dataset was morphologically and syntactically annotated (including POS tags, morphosyntactic features, and dependency relations; cf. Universal Dependencies, 2022d) by Google abiding to its universal annotation guidelines and converted to Universal Dependencies v2 guidelines by members pertaining to the Universal Dependencies community (cf. Zeman et al., 2021).

4.2 English Part of the Parallel Universal Dependencies (PUD)

This treebank consists of 21.183 tokens and 21.183 words in 1.000 English sentences. Annotation, format and license from 4.1 applies here, too (cf. Zeman et al., 2020; Universal Dependencies, 2022b).

4.3 GSD

The GSD treebank is "converted from the content head version of the universal dependency treebank v2.0" (Universal Dependencies, 2022c), the latter being a set of treebanks in multiple languages annotated in basic Standford-style dependencies. It is released under the CC BY-SA 4.0 license and consists of 15.590 German sentences originating from news, reviews, and Wikipedia comprising 287.725 tokens and 292.773 words in total (cf. Universal Dependencies, 2022e). The treebank is morphologically and syntactically annotated, including lemmas, POS tags, morphosyntactic features, and dependency relations (cf. Universal Dependencies, 2022c).

4.4 EWT

The EWT treebank in English covers weblogs, newsgroups, emails, reviews, and Yahoo! answers encompassing 251.528 tokens and 254.860 words in 16.622 sentences (cf. Universal Dependencies, 2022a). It is described as a "Gold Standard Universal Dependencies Corpus for English, built over the source material of the English Web Treebank LDC2012T13" (ibid.). Annotation-wise, it contains manual/automatic assignments of lemmas, POS tags, morphosyntactic features, and syntactic dependencies (cf. ibid.).

5 Evaluation Procedure

The following evaluation is primarily an automatic quantitative performance evaluation - i.e. a "measurement of system performance in one or more specific areas" (Hirschmann and Thompson, 1997, p. 410). The paradigm used to evaluate the CRF POS tagger, or rather its input-output characteristics, boils down to comparing the extent to which one model output (label) per input is correct or desirable, quantified by the use of metrics (cf. Resnik and Lin, 2010).

However, although the metrics are essential to measure the performance, correctness, and quality of a system, they are only an approximation of the system's qualities, as they are limited to their underlying criteria (cf. Thomas and Uminsky, 2022). Therefore, the report also includes a semi-automatic/manual linguistic error analysis to counteract their biases and drawbacks. This allows for a more in-depth evaluation of the performance of the system and to identify structures and patterns that are obscured by the numerical values of a metric.

The evaluation is not only summative, i.e. it is not carried out during but after the development of a system, but it is also a glass box evaluation due to the publicly available model implementation. However, further extrinsic evaluation would be required to provide a more global, representative, and non-isolated holistic view of the model. The impact of POS tagging could be diagnosed by performing an ablation study - "a set of experiments in which components of a machine learning system are removed/replaced in order to measure the impact of these components [...]" (Ali et al., 2023) - on the respective system in which it is embedded.

5.1 Confusion Matrix & Metrics

With the help of a confusion matrix, which compares the predicted labels/classes (here: POS-tags) of a system to the gold labels, different metrics can be computed globally or class-specifically by using the true/false positive counts of each class provided by the matrix (cf. Potts, 2022). The metrics used for the evaluation are: accuracy, precision, recall, and F1.

Accuracy scores represent the sum of the correct

predictions divided by the sum of all predictions, giving a value between 0 and 1, where 0 is the worst and 1 is the best. Put differently, it shows how often a classifier is globally correct, but does not provide per-class information. Furthermore, size imbalances contribute to a systematic bias and can lead to a misrepresentative accuracy score (cf. ibid.).

Precision is a per-class metric calculated as the sum of the correct predictions divided by the sum of all guesses for that class. It exhibits the same value range and positive directionality as accuracy. However, its weakness is that it gives high scores for classes exhibiting only few or no false positive predictions in contrast to the true positive predictions that lack many true instances (cf. ibid.).

Recall is a per-class metric that takes into account the true instances that were disregarded by precision: it is the sum of the correct predictions divided by the sum of all true instances. The range of values and positive directionality are the same as for the other metrics. As this metric focuses on the correct guesses, it exhibits a dangerous edge case when disregarding the incorrect ones, resulting in high recall values (cf. ibid.).

Finally, the F1 score provides a per-class trade-off between precision and recall by computing their harmonic mean. Missed true instances and incorrect predictions are thus both taken into consideration. Its value range (guaranteed to be between precision and recall) and positive directionality are consistent with the other metrics. The macro averaged F1 score is the F1 average over each class, but is prone to outliers.

5.2 Baselines

The purpose of baselines is to establish a benchmark for the performance of a system that is expected to either out- or underperform them. There are two categories of baselines employed in the evaluation: Firstly, a 'bad' general baseline devoid of knowledge needed for the respective task, applying a simple stochastic mechanism to classify data according to randomness or label frequency in the training data. In the evaluation, a dummy classifier is used that ignores the input features and makes random predictions based on the class distributions in the training data. Secondly, a 'good' baseline of some prior approach or a more sophisticated implementation that serves as the upper bound. The non-optimized CRF POS tagger is used as a good baseline in this evaluation, compared to its hyperparameter-optimized counterpart that is expected to perform better (cf. Resnik and Lin, 2010; Li et al., 2020).

5.3 Hyperparameter Optimization

Hyperparameter Optimization is an integral part of Machine Learning, exploring "the space of possible hyperparameter configurations systemtically, in a structured way, i.e. HPT is an optimization problem" (Bartz et al., 2023, p. 15). HPT is employed in the evaluation as an attempt to improve the performance of the aforementioned CRF POS tagger baseline (non-optimized one), i.e. to find a setup from a set of possible CRF configurations at training time that builds a better performing model.

5.4 K-Fold Cross Validation

To mitigate possible asymmetric distributions or clustering of (linguistic) phenomena in the data splits, k-fold cross validation is employed as an attempt to train and test over several disjoint data splits, thus producing more reliable and holistic scoring results that are less susceptible to "particularly fortuitous (or infortuitous) selection of test data" (Resnik and Lin, 2010, p. 278). The data is partitioned into k pieces or folds, thereby ensuring "that every item in the full data set gets used for both training and testing, while at the same time also ensuring that no item is used simultaneously for both purposes" (ibid., p. 279).

5.5 Linguistic Error Analysis

As outlined in the section on metrics, focusing on metrics alone can be problematic because their edge cases and weaknesses can mask and whitewash certain qualities and flaws of a model under the guise of numerical values. A linguistic error analysis that penetrates this cloak is essential to uncover what is hidden by the numbers through (semi-)automatically/manually analyzing the errors made by the model, thereby identifying patterns or common errors that provide valuable information not only about the real qualities of the model but also about possible ways to improve it.

6 Experiments

6.1 Good Baseline Model

Using the single training and test split of each dataset from 4, the non-hyperparameter-optimized CRF model from the `nltk.tag.crf` module is

trained and evaluated. The evaluation is performed with both the `classification_report` and `confusion_matrix` functions from the `sklearn.metrics` module, yielding results for all the (per-class) metrics from 5.2 and a plotted, storable confusion matrix (together with the class `ConfusionMatrixDisplay` from the metrics module). The default hyperparameters for the model are, using the LBFGS training algorithm (the only one available in NLTK): *'feature.min_freq'*: 0.0, *'feature.possible_states'*: False, *'feature.possible_transitions'*: False, *'c1'*: 0.0, *'c2': 1.0*, *'max_iterations'*: None, *'num_memories'*: 6, *'epsilon'*: 1e-5, *'period'*: 10, *'delta': 1e-5*, *'linesearch'*: 'MoreThuente', *'max_linesearch'*: 20

6.2 Poor Baseline Model

Using the `DummyClassifier` class from the `sklearn.dummy` module with its stratified strategy, all datasets are trained and evaluated on this baseline model. Training and evaluation is performed as outlined in the previous section on the good baseline.

6.3 Hyperparameter Tuning

For hyperparameter tuning, I did not use a pre-implemented algorithm such as `GridSearchCV` from `sklearn` but rather, for epistemic reasons, implemented a simple grid search algorithm myself, oriented towards the one proposed in Mueller and Guido, chapter 5.2.1 (2016), that searches exhaustively through a defined subset of the hyperparameter space. Given the amount of variable hyperparameters that easily mount up to several hundreds of thousands of combinations, I confined the search space as follows (1.944 combinations): *'c1'*: 0, 0.5, 1, *'c2'*: 1, 2, 3, *'max_iterations'*: None, 50, *'feature.possible_states'*: True, False, *'feature.possible_transitions'*: True, False, *'num_memories'*: 5, 6, 7, *'linesearch'*: 'MoreThuente', 'Backtracking', 'StrongBacktracking', *'max_linesearch'*: 10, 20, 30

The CRF model was trained on every hyperparameter combination using the training set from the GSD treebank, which is the largest dataset in this project. Testing at optimization time was performed on its validation set - so that the test split can be used for post-optimization testing - while keeping track of the currently best hyperparameter setting by computing the F1 score for each model configuration, using the `f1_score` function from the `sklearn.metrics` module. Finally, the best hyperparameter configuration was stored and used to train and test on every dataset as outlined in the previous sections. Explanations of the hyperparameters can be found here: NLTK CRF source code.

6.4 K-Fold Cross Validation

10-fold cross validation is performed on both PUD German and PUD English (the creators recommend 10 folds, cf. Zeman et al., 2021), both using the hyperparameter optimized model and its non-optimized baseline. As the hyperparameter optimization was conducted on a German dataset, this experiment will reveal how the optimization effects the performance in English and by means of the different data folds, to what extent the data is consistent in performance, meaning if the performance differs across folds due to possibly asymmetric distributions of (linguistic) phenomena. The performance consistency will be reflected by the standard deviation of the F1 score.

The `KFold` class from the `sklearn.model_selection` module is used to create the folds, yielding train and test indices for the respective fold. For each fold, evaluation is performed using the `classification_report` function from the `sklearn.metrics` module.

6.5 Linguistic Error Analysis

Because of its relatively small size facilitating a semi-automatic/manual analysis, the linguistic error analysis is conducted using the PUD English dataset and the non-optimized CRF to provide answers and information to/on the following aspects: error categorization (with which predicted labels was a gold label confused with), error frequency (frequency of different types of errors), error rate, type-token-ratio in the data splits and manual examination of 2 example sentences that were flawlessly/most poorly tagged, respectively.

7 Results

7.1 Non-Optimized CRF Model

Overall, the non-optimized CRF model performed well on all datasets with F1 values ranging from 0.78 (PUD ENG) to 0.87 (EWT). The German equivalent of the PUD ENG scores slightly better despite its equal size (F1 0.81). The GSD dataset is in fact the largest one, but not the one that performed best according to the F1 score - even the small PUD GER dataset performed slightly better and the second largest one, EWT, outperformed

them all. However, in terms of accuracy, the model does not exhibit large deviations across the datasets, amounting to a maximum of 0.02 points of difference with GSD and EWT performing globally best (0.91, respectively).

Dataset	Accuracy	Precision	Recall	F1
PUD GER	0.89	0.84	0.80	0.81
PUD ENG	0.88	0.82	0.76	0.78
GSD	0.91	0.88	0.79	0.8
EWT	0.91	0.89	0.86	0.87

Table 1: Accuracy, precision, recall, and F1 score values of the CRF baseline model (non-optimized) on four different datasets. Values for precision, recall, and F1 are macro averaged.

7.2 DummyClassifier

As expected, the `DummyClassifier` performs rather poorly on all datasets, given the mere number of possible labels (POS tags; 16-18, depending on the dataset) that can be assigned to a token. All scores are relatively similar, with little variation across the datasets, reaching a maximum of 0.1 (accuracy) and 0.06 (precision, recall, and F1, respectively). Therefore, this baseline model is unsuitable for real-world applications.

Dataset	Accuracy	Precision	Recall	F1
PUD GER	0.1	0.06	0.06	0.06
PUD ENG	0.1	0.06	0.06	0.06
GSD	0.1	0.05	0.05	0.05
EWT	0.09	0.05	0.05	0.05

Table 2: Accuracy, precision, recall, and F1 score values of the poor baseline model (DummyClassifier) on four different datasets. Values for precision, recall, and F1 are macro averaged.

7.3 Optimized CRF Model

Hyperparameter optimization yielded the following parameters, maximizing the macro averaged F1 score at training time:

'c1': 0.5, *'c2'*: 1, *'max_iterations'*: 50, *'feature.possible_states'*: True, *'feature.possible_transitions'*: False, *'feature.minfreq'*: 0, *'num_memories'*: 6, *'delta'*: 0.001, *'linesearch'*: *'MoreThuente'*, *'max_linesearch'*: 10, *'epsilon'*: 0.0001

In comparison to the good baseline, i.e. the non-optimized CRF model, hyperparameter tuning as performed in this project resulted only in a slight improvement of the F1 scores (0.01 for each dataset, except for GSD whose performance stagnated). The differences in accuracy, precision, and recall are also negligible as they only amount to 0.01 positive/negative points. It indicates that the tuning was not thoroughly enough conducted and that the search space might have been too confined or unfavorably selected.

Dataset	Accuracy	Precision	Recall	F1
PUD GER	0.88	0.84	0.81	0.82
PUD ENG	0.87	0.82	0.77	0.79
GSD	0.91	0.88	0.78	0.8
EWT	0.91	0.89	0.87	0.88

Table 3: Accuracy, precision, recall, and F1 score values of the hyperparameter-optimized CRF model on four different datasets. Values for precision, recall, and F1 are macro averaged.

7.4 K-Fold Cross Validation

7.4.1 Optimized Model

The metric values indicate that the optimized CRF model performs only slightly better on the PUD ENG dataset in 10-fold cross validation compared to the non-optimized train test procedure (F1 of 0.829 vs. 0.82) . These negligible performance differences suggest that the data in PUD GER is relatively evenly distributed, mirrored in the low F1 standard deviation of 0.023 (second lowest among the 4 experiments).

Metric	Value
Accuracy	0.901
Precision	0.839
Recall	0.823
F1	0.829
F1 standard deviation	0.023

Table 4: Accuracy, precision, recall, and F1 score values of the hyperparameter-optimized CRF model on the PUD GER dataset, averaged over 10 folds. Values for precision, recall, and F1 are macro averaged. F1 standard deviation is calculated over 10 folds.

As for 10-fold cross validation on the PUD ENG dataset, the F1 standard deviation of 0.043, representing the highest among the four experiments, together with an improved F1 score of 0.828 (10-fold) compared to 0.79 (standard train-test non-optimized), indicates that the data is less evenly distributed in the PUD ENG dataset. The appendix provides a plot of the F1 scores over all folds for this experiment, showing that especially after the second and eighth fold, the performance varies abruptly between the highest F1 score of 0.88 (fold 7) and the lowest one of 0.75 (fold 9), while being

relatively stable at the beginning (fold 1-2, 4) and in the middle (fold 5-7).

Metric	Value
Accuracy	0.886
Precision	0.847
Recall	0.817
F1	0.828
F1 standard deviation	0.043

Table 5: Accuracy, precision, recall, and F1 score values of the hyperparameter-optimized CRF model on the PUD ENG dataset, averaged over 10 folds. Values for precision, recall, and F1 are macro averaged. F1 standard deviation is calculated over 10 folds.

7.4.2 Non-Optimized Model

In terms of F1 score, the non-optimized CRF model performed slightly better during 10-fold cross validation on the PUD GER dataset compared to the standard train-test procedure (non-optimized model), and neglectably poorer than during 10-fold cross validation with the optimized CRF model (F1s of 0.828, 0.81, and 0.829, respectively). This indicates that the optimization enhanced the model performance only subtly for this experiment and that the PUD GER dataset is relatively balanced in terms of linguistic phenomena - the German datasets exhibit both lower F1 standard variations compared to their English counterpart. However, this experiment yielded the highest accuracy score out of the three mentioned experiments, amounting to 0.902 and thus indicating that it performs globally best if no metric bias is assumed - compared to the 10-fold cross validation using the optimized model (accuracy of 0.901) and the standard train-test procedure (accuracy of 0.89).

Metric	Value
Accuracy	0.902
Precision	0.834
Recall	0.82
F1	0.828
F1 standard deviation	0.022

Table 6: Accuracy, precision, recall, and F1 score values of the non-optimized CRF model on the PUD GER dataset, averaged over 10 folds. Values for precision, recall, and F1 are macro averaged. F1 standard deviation is calculated over 10 folds.

The non-optimized CRF model performed slightly better on the PUD ENG dataset (F1 0.83) than the optimized one during 10-fold cross validation (F1 0.828) and the non-optimized during the train-test procedure (0.78), suggesting that the optimization did not result in a representative model improvement for this experiment but that the PUD ENG dataset might be more unbalanced in terms of linguistic phenomena, mirroring in the enhanced F1 score when the data is randomly distributed in ten folds. Furthermore, the accuracy score of 0.849 in this experiment compared to 0.88 and 0.886 in the optimized 10-fold cross validation and non-optimized train-test procedure also indicates that the optimization was not as profitable as expected and that the foldings reproduced a subtle data unbalance.

Metric	Value
Accuracy	0.849
Precision	0.849
Recall	0.812
F1	0.830
F1 standard deviation	0.039

Table 7: Accuracy, precision, recall, and F1 score values of the non-optimized CRF model on the PUD ENG dataset, averaged over 10 folds. Values for precision, recall, and F1 are macro averaged. F1 standard deviation is calculated over 10 folds.

7.5 Linguistic Error Analysis

Even though the results from the train-test procedure on the PUD ENG dataset using the non-optimized CRF model indicate a good performance, the linguistic error analysis was able to unveil some of the model's flaws. Out of the 200 test sentences, only 35 were entirely correctly POS-tagged. 165 sentences exhibited at least one wrongly assigned POS token, thus accounting for an error rate of approximately 0.79 on sentence level. The average number of wrong predictions per sentence amounts to approximately 2.5.

Out of 4169 gold labels in the test sentences, 505 were wrongly assigned by the model, amounting to a label error rate of approximately 0.12. Among the labels, the 10 most common wrongly predicted ones paired with the respective tokens and counts are: (('to', 'PART'), 11) (('to', 'ADP'), 9) (('that', 'SCONJ'), 8) (('%', 'PUNCT'), 7) (('as', 'ADP'), 6) (('more', 'ADJ'), 5) (('not', 'PART'), 5) (('have', 'AUX'), 4) (('is', 'AUX'), 4) (('me', 'NOUN'), 3). This indicates that even allegedly trivial-seeming tokens such as 'to', 'that', and 'me' might be difficult to distinguish for the model, yet they are important for accurate natural language processing. On the other hand, the 10 most com-

mon wrongly predicted labels with counts only are: ('NOUN', 134) ('PROPN', 85) ('VERB', 67) ('ADJ', 58) ('ADP', 50) ('ADV', 31) ('PART', 21) ('AUX', 19) ('DET', 12) ('SCONJ', 10), suggesting that the model is more likely to mislabel certain parts of speech than others, especially nouns and proper nouns, possibly due to similarities in their syntactic or semantic properties and particularly if embedded in complex constructions.

The confusion matrix (cf. appendix) reveals that the class 'NOUN', being the most confused one, was most often mislabeled as 'PROPN' and 'VERB', indicating that the recognition of proper nouns/named entity recognition has its flaws (also in the opposite direction, 'PROPN' being labeled as 'NOUN'). Furthermore, the 'NOUN'-'VERB' confusion might be due to the feature function used in the CRF model that leads to a misinterpretation of nouns with '-ing' suffixes as verbs.

The sentences number 45 and 198 (cf. appendix) exhibit the most mislabels, both amounting to 10 errors. Number 45 contains a lot of proper nouns that the model did not label correctly which is due to both a song and album title that span over several tokens ('Her Father Didn't Like Me Anyway', 'Over My Head') that were not recognized as an associated proper noun. Sentence 198 exhibits mislabels particularly pertaining to subordinating conjunctions ('SCONJ' label). For example, the CRF model was not able to label the conjunction 'prior to' correctly as a related entity - both 'prior' and 'to' should be labeled as 'SCONJ', but the model labeled them as 'PROPN' and 'PART', respectively. This provides further evidence, together with the findings on the proper nouns in sentence number 45, that the model has difficulties when it comes to token spans that constitute a related entity.

The sentences 191 and 200 are two examples out of the 35 sentences that were flawlessly labeled by the model. Both are not only shorter than sentence 45 and 198, but especially sentence 191 is syntactically simple and contains several tokens that are expected to be easily labeled, such as numbers and punctuation marks. However, all proper nouns in sentence 45 were correctly labeled as such (4 proper nouns out of 19 tokens in total). Sentence 200 also exhibits a rather easy syntactic constitution that probably contributed to the flawless model predictions, next to the high frequency tokens and multi-word expressions such as 'an example of'.

Analyzing the type-token ratio (TTR) in the training and test split yielded values of approximately 30 and 44.5, respectively. The value mirrors the token variability in the split, where a high TTR represents a high lexical variation while a low one indicates the opposite. The TTR difference between the splits indicates, as already implied by the results of the 10-fold cross validation on the PUD ENG dataset, that the data is not evenly distributed. Furthermore, the higher value in the test split might account for poorer labeling results because a high variability might also entail sparse data for infrequently occuring tokens.

8 Conclusion

The transparent foundation for the different experiments was laid through a comprehensive introduction to the POS tagging task, the CRF model, and the datasets followed by an elaboration on the evaluation procedure. Using 4 different datasets, 3 different models - 2 baselines and a hyperparameter-optimized one as the upper bound - , k-fold cross validation, and a linguistic error analysis, the CRF model was examined from various perspectives conducting a wide variety of common procedures in NLP.

While the poor baseline model yielded rather bad results during the simple train-test procedure with F1 values peaking at 0.06, the non-optimized CRF model performed well on all datasets and reached a maximum F1 value of 0.87 on the EWT dataset. In 10-fold cross validation, however, this maximum could not be reached by the non-optimized model, maybe because it was performed on the small PUD GER and PUD ENG datasets. Yet on these two datasets, the 10-fold cross validation showed an improvement compared to the single train-test split in terms of F1 values, providing evidence that the folding might result in a better distribution of the training and test data where the effect of '(un-)lucky' splits is mitigated.

Contrary to expectations, the hyperparameter tuning of the CRF model was not as effective as desired and mirrored, if only, in slight model improvements that can be neglected. However, this is a starting point for further model improvements: a more thoroughly conducted hyperparameter optimization with a larger and more diverse search space can probably enhance the model performance, but it is rather time consuming.

The linguistic error analysis yielded that the model has its difficulties with longer sentences with

a lot of proper nouns, low frequency tokens/multi-word expressions, and syntactically more complex structures. It's important to note that while a label error rate of 0.12 may seem small, it can still have significant implications for the accuracy of the model's predictions in real-world applications. Therefore, ongoing evaluation and improvement of models is crucial to ensure their effectiveness and reliability. Improvements might involve some named entity recognition functionality and generally an improved dataset. The latter should come from more diverse sources with an appropriate size - it was shown that dataset size alone is not the only factor - also being very balanced and covering a wide variety of linguistic phenomena, including named entities, complex syntactic structures, and low frequency tokens/expressions. Furthermore, considering a more balanced type-token-ratio in the training and test data might also enhance the model's performance, as well as the adaption of the feature function that the CRF model uses.

My personal takeaways from this project are not only of practical, but also of theoretical nature. I learned to engage in the relevant literature and Python libraries and to extract ideas and concepts that I then put into practice. Working with the CRF model, I gained a deeper understanding of its implementation and its application to part-of-speech tagging through feature engineering. Furthermore, I acquired valuable experience in data preparation and evaluation procedures, including how data is split into training and test sets (or even in k-folds), how to design experiments, as well as how to interpret, communicate, and visualize results. The project also made me more aware of the practical challenges involved in developing and evaluating a model.

9 Responsible NLP Considerations

After the release of the chatbot 'ChatGPT' that at present not only baffles and dominates, but also divides the (AI) world, ethical considerations in NLP are now more important and pervasive than ever. Even though POS tagging might seem to be a trivial (preprocessing) task at first sight, its integration into larger systems can turn it into an accomplice if the system is exploited in a harmful way. For instance, dual use for nefarious purposes such as targeted harassment or surveillance poses a serious threat not only to society but also to individuals. This underlines the need for responsible and careful production and use of NLP systems.

Another important consideration is the ethical collection and use of data. The dataset(s) used to train a model should always be ethically sourced and properly licensed, and any potential biases in the data should be addressed to ensure that the model produces fair and accurate results. For instance, there may be demographic biases (exclusion or demographic misrepresentation of certain groups) or topic under-/overexposure that can significantly affect the performance of a model. In this report, I have been careful to use only licensed datasets from a trusted source. In addition, the results and procedures used in this project have been carefully presented to ensure transparency, explainability, and accountability - aspects that are often neglected in AI research and publication that might mislead the reader and lead to misinterpretation if disregarded.

In the face of the replication crisis, a troubling trend in AI that hinders the reproducibility of a model or results, I have tried my best to establish clarity and openness. For the sake of knowledge and truth, it is important to ensure that models/results can be replicated, not only by transparent communication but also, if possible, by providing the respective source code the model/results are based on. This is crucial for external verification and validation by other researchers, ensuring that the field of NLP continues to advance and produce reliable, high-quality models.

Finally, as mentioned in the section on the metrics, a common problem in NLP is the centering on or overemphasis on mere metric values. This can lead to "real-world harms, including manipulation, gaming, and a myopic focus on short-term qualities and inadequate proxies" (Thomas and Uminsky, 2022, p. 1) . This effect was mitigated not only by conducting a wide variety of experiments that attempted to uncover the underlying qualities and flaws of the CRF model, but also by providing a linguistic error analysis that penetrated the guise of metrics.

In summary, ethical considerations are a critical component of any NLP project, even for small systems such as the CRF model used for POS tagging. By carefully considering the factors mentioned - and they do not represent the entirety -, researchers and practitioners can ensure that their work is responsible, ethical, and holistically beneficial to society.

References

Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. 2023. Running an Ablation Study. Last Accessed: 2023-03-26.

Eva Bartz, Thomas Bartz-Beielstein, Martin Zaefferer, and Olaf Mersmann, editors. 2023. *Hyperparameter Tuning for Machine and Deep Learning with R - A Practical Guide*. Springer.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc.

Alebachew Chiche and Betselot Yitagesu. 2022. Part of Speech Tagging: A systematic Review of Deep Learning and Machine Learning Approaches. *Journal of Big Data*, 9(1):10.

Paul R. Cohen and Adele E. Howe. 1988. How Evaluation Guides AI Research: The Message Still Counts More than the Medium. *AI Magazine*, 9(4):35.

Lynette Hirschmann and Henry S. Thompson. 1997. Evaluation. In Ron Cole et al., editors, *Survey of the State of the Art in Human Language Technology*, page 409–438. Cambridge University Press.

Daniel Jurafsky and James H. Martin. 2020. *Speech and Language Processing (3rd Edition draft)*.

Margaret King. 1996. Evaluating Natural Language Processing Systems. *Commun. ACM*, 39(1):73–79.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Dennis Li, Euxhen Hasanaj, and Shuo Li. 2020. 3 - Baselines. Last Accessed: 2023-04-05.

A.C. Müller and S. Guido. 2016. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Christopher Potts. 2022. Evaluation Metrics in NLP. Last Accessed: 2023-04-05.

Philip Resnik and Jimmy Lin. 2010. Evaluation of nlp systems. In *The Handbook of Computational Linguistics and Natural Language Processing*, chapter 11, pages 271–295. John Wiley Sons, Ltd.

Bhargav Srinivasa-Desikan. 2018. POS-Tagging and Its Applications. In *Natural Language Processing and Computational Linguistics*. Packt Publishing, Limited, United Kingdom.

Rachel L. Thomas and David Uminsky. 2022. Reliance on Metrics is a Fundamental Challenge for AI. *Patterns*, 3(5):1–8.

Universal Dependencies. 2022a. UD English EWT. Last Accessed: 2023-03-25.

Universal Dependencies. 2022b. UD English PUD. Last Accessed: 2023-03-24.

Universal Dependencies. 2022c. UD German GSD. Last Accessed: 2023-03-25.

Universal Dependencies. 2022d. UD German PUD. Last Accessed: 2023-03-24.

Universal Dependencies. 2022e. Universal Dependencies. Last Accessed: 2023-03-23.

Daniel Zeman et al. 2020. Universal Dependencies - UD English PUD. Last Accessed: 2023-03-24.

Daniel Zeman et al. 2021. Universal Dependencies - UD German PUD. Last Accessed: 2023-03-24.

Daniel Zeman et al. 2022. Universal dependencies 2.11. LINDAT/CLARIAH-CZ Digital Library at the Institute of Formal and Applied Linguistics (UFAL), Faculty of Mathematics and Physics, Charles University.

Appendix

10-Fold Cross Validation Plot

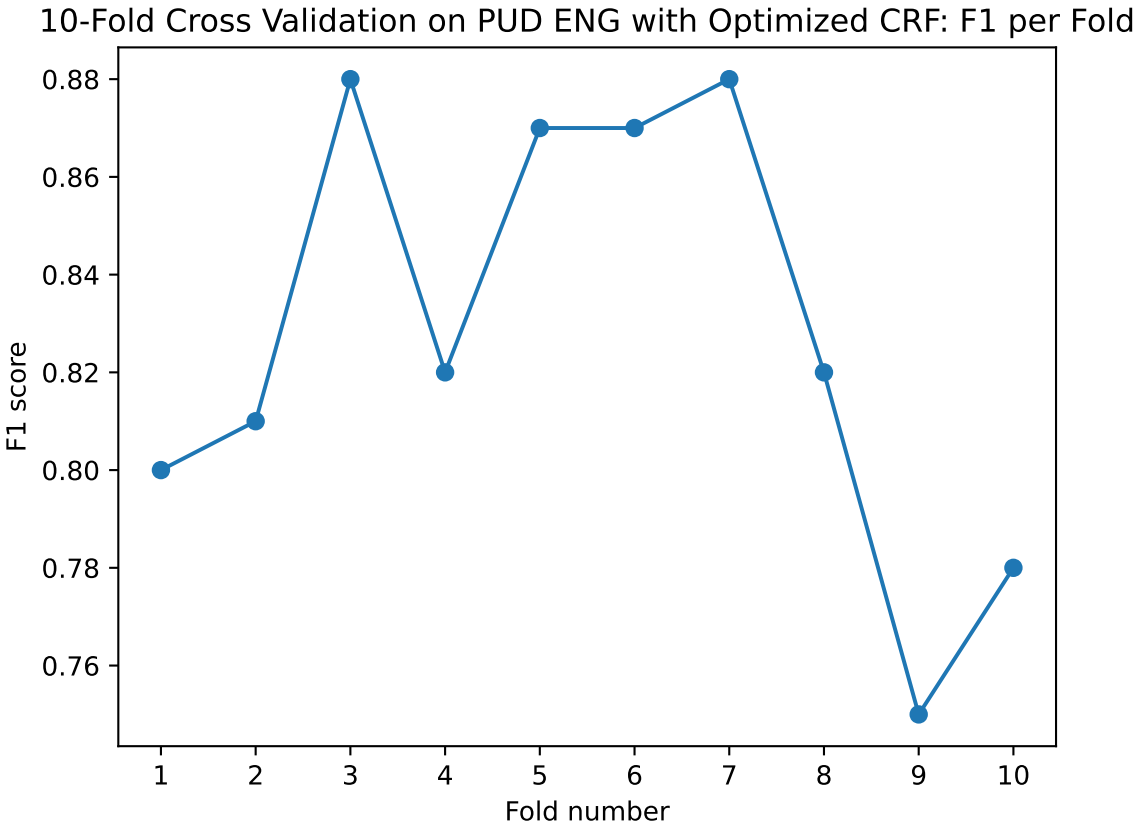

Figure 2: F1 score (macro averaged) per evaluated fold during 10-fold cross validation on the PUD ENG dataset, using the optimized CRF model. Standard deviation for the F1 score amounts to 0.043, which is the highest among the 4 10-fold cross validation experiments.

Confusion Matrix

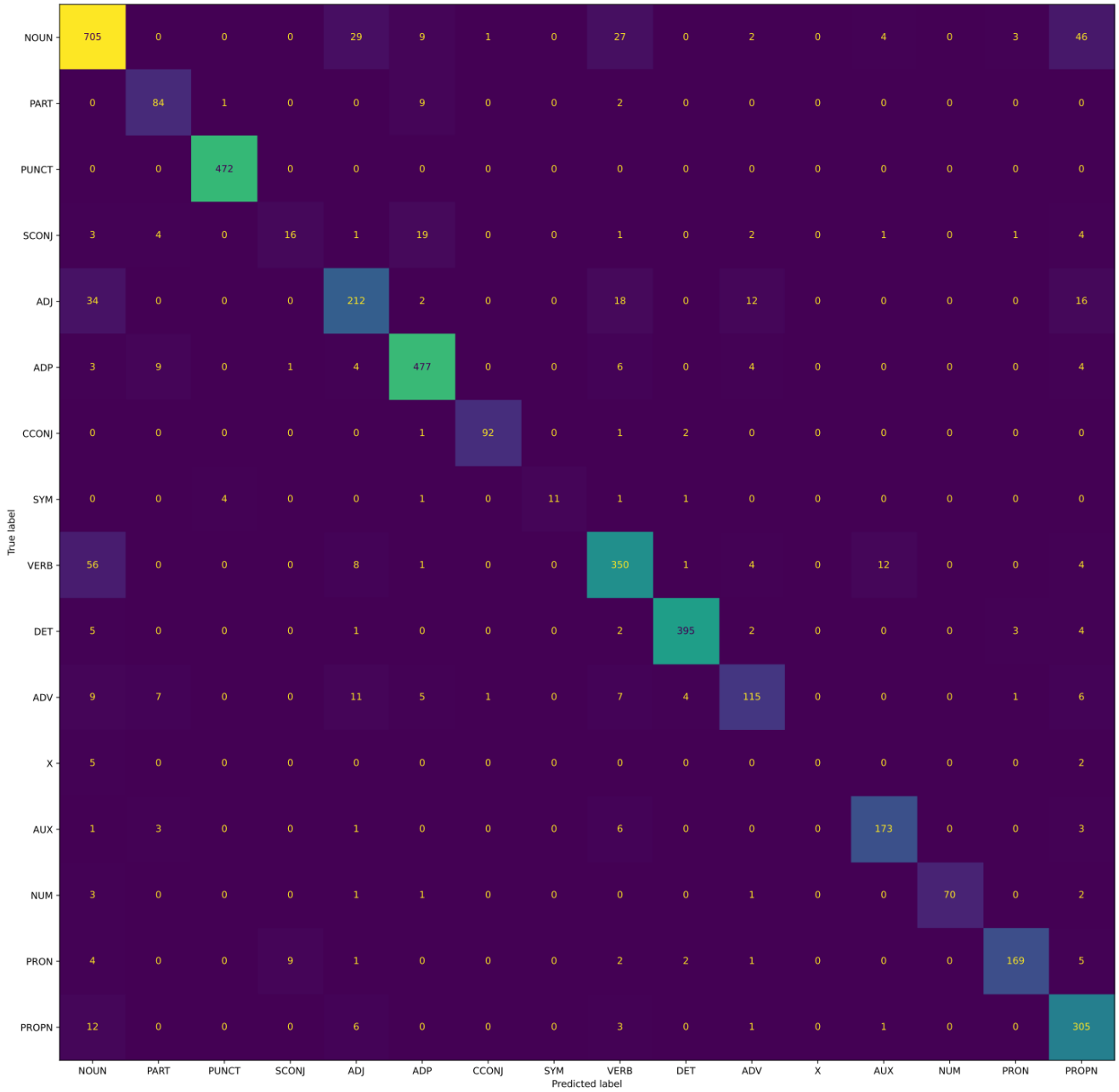

Figure 3: Confusion Matrix of the non-optimized CRF tagger on the PUD ENG dataset.

Example Sentences for Linguistic Error Analysis

The following sentences pertain to the PUD ENG test set. Structure: Model prediction per token is the first element of each tuple, whereas the second element represents the gold label.

Sentences with the Most Errors

Sentence 45 (44 index, 10 errors): [(('Rafferty', 'PROPN'), ('Rafferty', 'PROPN')), (('recorded', 'VERB'), ('recorded', 'VERB')), (('a', 'DET'), ('a', 'DET')), (('new', 'ADJ'), ('new', 'ADJ')), (('version', 'NOUN'), ('version', 'NOUN')), (('of', 'ADP'), ('of', 'ADP')), (('his', 'PRON'), ('his', 'PRON')), (('Humblebums', 'NOUN'), ('Humblebums', 'PROPN')), (('song', 'NOUN'), ('song', 'NOUN')), (('"', 'PUNCT'), ('"', 'PUNCT')), (('Her', 'PROPN'), ('Her', 'PRON')), (('Father', 'PROPN'), ('Father', 'NOUN')), (('Did', 'PROPN'), ('Did', 'AUX')), (("n't", 'PART'), ("n't", 'PART')), (('Like', 'PROPN'), ('Like', 'VERB')), (('Me', 'PROPN'), ('Me', 'PRON')), (('Anyway', 'PROPN'), ('Anyway', 'ADV')), (('"', 'PUNCT'), ('"', 'PUNCT')), (('on', 'ADP'), ('on', 'ADP')), (('the', 'DET'), ('the', 'DET')), (('album', 'NOUN'), ('album', 'NOUN')), (('Over', 'PROPN'), ('Over', 'ADP')), (('My', 'PROPN'), ('My', 'PRON')), (('Head', 'PROPN'), ('Head', 'NOUN')), (('(', 'PUNCT'), ('(', 'PUNCT')), (('1994', 'NUM'), ('1994', 'NUM')), ((')', 'PUNCT'), (')', 'PUNCT')), (('.', 'PUNCT'), ('.', 'PUNCT'))]

Sentence 198 (197 index, 10 errors): [(('Prior', 'PROPN'), ('Prior', 'SCONJ')), (('to', 'PART'), ('to', 'SCONJ')), (('taking', 'VERB'), ('taking', 'VERB')), (('office', 'NOUN'), ('office', 'NOUN')), (('Jokowi', 'PROPN'), ('Jokowi', 'PROPN')), (('sought', 'VERB'), ('sought', 'VERB')), (('for', 'SCONJ'), ('for', 'ADP')), (('outgoing', 'VERB'), ('outgoing', 'ADJ')), (('President', 'PROPN'), ('President', 'PROPN')), (('Yudhoyono', 'PROPN'), ('Yudhoyono', 'PROPN')), (('to', 'PART'), ('to', 'PART')), (('take', 'VERB'), ('take', 'VERB')), (('responsibility', 'NOUN'), ('responsibility', 'NOUN')), (('for', 'ADP'), ('for', 'ADP')), (('the', 'DET'), ('the', 'DET')), (('decision', 'NOUN'), ('decision', 'NOUN')), (('to', 'ADP'), ('to', 'PART')), (('further', 'ADJ'), ('further', 'ADV')), (('increase', 'NOUN'), ('increase', 'VERB')), (('fuel', 'NOUN'), ('fuel', 'NOUN')), (('prices', 'NOUN'), ('prices', 'NOUN')), (('by', 'ADP'), ('by', 'SCONJ')), (('further', 'ADJ'), ('further', 'ADV')), (('removing', 'NOUN'), ('removing', 'VERB')), (('subsidies', 'NOUN'), ('subsidies', 'NOUN')), (('.', 'PUNCT'), ('.', 'PUNCT'))]

Perfectly Labeled Sentences

Sentence 191 (190 index, 0 errors): [(('In', 'ADP'), ('In', 'ADP')), (('1912', 'NUM'), ('1912', 'NUM')), (('was', 'AUX'), ('was', 'AUX')), (('founded', 'VERB'), ('founded', 'VERB')), (('the', 'DET'), ('the', 'DET')), (('first', 'ADJ'), ('first', 'ADJ')), (('film', 'NOUN'), ('film', 'NOUN')), (('company', 'NOUN'), ('company', 'NOUN')), (('(', 'PUNCT'), ('(', 'PUNCT')), (('Athina', 'PROPN'), ('Athina', 'PROPN')), (('Film', 'PROPN'), ('Film', 'PROPN')), ((')', 'PUNCT'), (')', 'PUNCT')), (('and', 'CCONJ'), ('and', 'CCONJ')), (('in', 'ADP'), ('in', 'ADP')), (('1916', 'NUM'), ('1916', 'NUM')), (('the', 'DET'), ('the', 'DET')), (('Asty', 'PROPN'), ('Asty', 'PROPN')), (('Film', 'PROPN'), ('Film', 'PROPN')), (('.', 'PUNCT'), ('.', 'PUNCT'))]

Sentence 200 (199 index, 0 errors): [(('An', 'DET'), ('An', 'DET')), (('example', 'NOUN'), ('example', 'NOUN')), (('of', 'ADP'), ('of', 'ADP')), (('a', 'DET'), ('a', 'DET')), (('desert', 'NOUN'), ('desert', 'NOUN')), (('island', 'NOUN'), ('island', 'NOUN')), (('would', 'AUX'), ('would', 'AUX')), (('be', 'AUX'), ('be', 'AUX')), (('the', 'DET'), ('the', 'DET')), (('small', 'ADJ'), ('small', 'ADJ')), (('islands', 'NOUN'), ('islands', 'NOUN')), (('off', 'ADP'), ('off', 'ADP')), (('the', 'DET'), ('the', 'DET')), (('coast', 'NOUN'), ('coast', 'NOUN')), (('of', 'ADP'), ('of', 'ADP')), (('Baja', 'PROPN'), ('Baja', 'PROPN')), (('California', 'PROPN'), ('California', 'PROPN')), ((',', 'PUNCT'), (',', 'PUNCT')), (('Mexico', 'PROPN'), ('Mexico', 'PROPN')), (('.', 'PUNCT'), ('.', 'PUNCT'))]