# COMSM0089 Introduction to Data Analytics - Task 2

## 1 Tagging

The task of Named Entity Recognition (NER) involves identifying and classifying text spans, referred to as "tokens", that correspond to specific named entities (anything with a proper name). These tokens are then assigned an appropriate "label". Common labels used in NER include PER (person), LOC (location), and ORG (organisation). This task differs from part-of-speech (POS) tagging, wherein each word, or "token", is given a "tag" representing its grammatical role. The NER task is more complex as it requires understanding the boundaries of entities that may span multiple words. The BioNLP 2004 dataset, for instance, includes ten unique labels representing entities such as DNA, protein, cell type, cell line, and RNA.

A Conditional Random Field (CRF) model is trained and evaluated for generalised performance to implement a method for labelling entities.

### 1.1 Methodology

CRF is "a discriminative sequence model based on log-linear models" (Jurafsky and Martin, 2019, p. 174, Ch. 6) and is a probabilistic model, similar to Logistic Regression, but designed for sequences of tokens, not just individual ones. CRFs use functions to generate "feature vectors" for each token in a sequence. In combination with the labels of the neighbouring tokens, the CRF uses these feature vectors to predict a label for each token. Predictions are made dependently—the CRF computes the most likely sequence of labels for the entire sequence at once using an iterative approach based on the Viterbi algorithm (Forney, 1973). Starting from the first token in a sequence, it calculates the probability of each potential label being the correct one based on the feature vector of that token. This is repeated for each token in the sequence while keeping track of the most likely labels (those with the maximum probability). Once it has done this for all tokens and all possible labels, it traces back through the labels with the highest probabilities to identify the most likely sequence of labels.

#### 1.1.1 Strengths & Limitations

- Sequence Understanding: the CRF considers context and meaning from the whole sentence. Context dependency is beneficial because the meaning of words can significantly change based on their surroundings, e.g. "New York" [LOC] has a different entity tag when considered part of "New York Police Department" [ORG].
- Local Feature Functions: can be tailored to suit the application and are discussed in Section 1.4. Performance can suffer if features are not informative. Developing and testing feature sets can be time-consuming and complex.
- Discriminative Training: CRFs model the conditional probability of labels given the inputs, which allow them to focus on the specific task, potentially leading to improved performance. However, this also means the CRF does not support transfer learning, so they cannot leverage pre-trained models and must learn from scratch.
- Expense: CRFs can be slow to train, especially on large datasets where the algorithm may require a lot of computational resources.

### 1.2 Encoding entity spans

Entity spans in the text data are encoded as tags for each token using the "BIO" tagging scheme: "B", "I", and "O". "B" indicates the beginning, "I" signifies the interior, and "O" represents a token outside any entity. Each token in a sequence receives a tag that identifies whether it is part of an entity and its position within it if it is part of one. This method allows the model to recognise and differentiate entities even when they span multiple tokens. For example, the following sentence is shown with its predicted NER tags:

Number of glucocorticoid receptors in lymphocytes and their sensitivity to hormone action

Number/O of/O glucocorticoid/B-protein receptors/I-protein in/O lymphocytes/B-cell_type and/O their/O sensitivity/O to/O hormone/O action/O ./O

### 1.3 Software implementation

Python version: 3.9.16 (conda-forge) on a Windows 10 platform utilising sklearn and nltk python packages.

### 1.4 Local Feature Functions

The following features provide the model with different semantic and syntactic meanings, enabling it to make more accurate predictions. These features should impact results positively by covering many attributes that can hint at whether a word is an entity and what kind it might be. A custom CRF tagger captures various features, including:

- Position: Current, previous and following word for the context. Words surrounding the current word are critical in determining the entity of that word.
- Capitalisation: whether the first letter of a word is capitalised or not. Capitalisation can signify proper nouns in English (and many other languages).
- Number: if a token contains a numerical digit, this may indicate a specific type of entity, for example, a measurement or number label.
- Punctuation: check if a token is made of punctuation characters, making it unlikely to be an entity token.
- Suffixes and prefixes: characters from a word's beginning or end can hint at its grammatical meaning or role. For example, in English, many plurals end in "s", and past tense verbs end in "ed", helping identification.
- Word Shape: when dealing with unknown words, "One of the most important is word shape features" (Jurafsky and Martin, 2019, p. 176). Word Shape represents the letter pattern of a word by mapping lower case to "x", upper case to "X" and numbers to "d". WordShape provides a generic representation for words with similar patterns, which can be helpful if that exact word is not in the training data.
- Parts Of Speech (POS) Tags: Words with certain POS tags (like proper nouns) are more likely to be named entities and can help distinguish between similar words used as verbs or nouns in different contexts.

## 2 Evaluating and interpreting results

### 2.1 Metrics

F1 Score is widely used for evaluating NER and is defined as the harmonic mean of precision and recall. Precision is the ratio of true positive predictions (i.e. correct entity identifications) out of all positive predictions. Recall is the ratio of true positive predictions to all actual positive instances.

#### 2.1.1 Limitations

- The F1 score can be misleading in the case of imbalanced data, as it can be high by simply predicting the majority class. It is sensitive to Type I (false positive) and Type II (false negative) errors. A perfect F1 score does not necessarily equate to perfect model performance.
- Calculating F1 per class can obscure overall performance, making a macro-average (which computes the metric for each class and then takes the average) useful for a comprehensive evaluation.
- F1 score treats all entities as equally important, which may only sometimes be practical; for example, if we want to find people [PER], other entities such as places and organisations may not be valuable.

### 2.2 Test Procedure

The model was trained iteratively in three phases on the training set, with each phase introducing additional features as hyperparameters. The performance was evaluated based on a 'devset' after each phase, allowing for continuous improvement. PHASE 1 trained the default CRF tagger (NLTK Team, 2023a). PHASE 2 utilised a custom CRF tagger implementing the features outlined in Section 1.4. PHASE 3 implemented the custom CRF tagger from phase 2, augmented with POS tags (NLTK Team, 2023b). FINAL TEST, the final model performance is evaluated using a separate test set. This 'held-out' test set provides a means to determine how our model generalises to unseen data, more realistically estimating performance in a real-world setting.

### 2.3 Results

The impact of the different phases on the devset and test set and the time to train the models is shown in Table 1. More complex models took longer to train but achieved better performance. In PHASE2, we see an evident boost in the F1 scores across all classes, possibly due to the more complex model or additional features introduced during this phase. In PHASE3, although the training time increased further, there were no significant improvements in the F1 scores, which could suggest overfitting.

Table 1: F1 Scores and Training Time Results

| PHASE | DNA | protein | cell_type | cell_line | RNA | macro_F1 | t |
|---|---|---|---|---|---|---|---|
| PHASE1 | 0.58 | 0.73 | 0.63 | 0.54 | 0.65 | 0.63 | 110.46s |
| PHASE2 | 0.67 | 0.79 | 0.76 | 0.66 | 0.67 | 0.71 | 213.6s |
| PHASE3 | 0.67 | 0.79 | 0.76 | 0.67 | 0.65 | 0.71 | 262.06s |
| FINAL TEST | 0.70 | 0.71 | 0.64 | 0.55 | 0.62 | 0.65 | 262.06s |

### 2.4 Potential Improvements

- Using domain-specific POS tagging, as NLTK POS tagging may not be optimal for this dataset.
- Tuning the regularisation parameter for the CRF model to avoid potential overfitting.
- Data augmentation may help the model address its underperformance on certain classes, for example, cell_line.
- Incorporating advanced word embeddings like GloVe or BERT, though these may increase computation time.
- Implementing k-fold cross-validation to estimate confidence intervals for prediction.
- Exploring different models like transformer architectures.

## References

G. D. Forney. 1973. The viterbi algorithm. *Proc. of the IEEE*, 61:268 – 278, March.

Daniel Jurafsky and James H. Martin. 2019. *Speech and Language Processing*. Draft Version, 3rd edition. Online.