Lab Deliverable 2: NERC Task

Javier Herrer Torres
javier.herrer.torres@estudiantat.upc.edu

Ignacio Lloret Lorente
ignacio.lloret@estudiantat.upc.edu

1 Introduction

This report presents an exploration into Named Entity Recognition and Classification (NERC) tasks using machine learning techniques. The primary goal of this work is to develop a system capable of identifying and categorizing drug names within unstructured text data.

The project focuses on implementing a Sequence Tagging approach using the B-I-O schema.

To accomplish the NERC task, the project adopts a structured approach consisting of feature extraction, model training, and classification. Feature extraction involves capturing relevant information about tokens and their contexts to inform the learning process. Various machine learning algorithms are explored for model training, including Conditional Random Fields (CRF) and Naive Bayes, with an emphasis on experimenting with different parameterizations and feature combinations to optimize performance.

For this, we will play with different features extracted from the tokens in order to accomplish this task. We will also try different algorithms to try to improve the CRF model. We will measure the improvements made by the feature extractor by predicting a test dataset containing entities that the model must predict its category. As the main metric, we will use the recall and F1 in order to measure how the model performs in an unbalanced target challenge.

The deliverables of this project include a comprehensive report detailing the methodologies employed and insights gained from the experimentation process. Additionally, the report includes the extract features function, which forms a critical component of the feature extraction process, along with any relevant code snippets.

Overall, the objective of this project is to develop a proficient NERC system for drug name identification and classification, leveraging machine learning techniques to achieve accurate and efficient entity recognition within unstructured text data and test the importance of feature extraction in the context of NERC.

2 Machine Learning NERC

After experimenting with multiple algorithms including Gradient Boosted Trees, Multinomial Naive Bayes, and Random Forest, we found that none of them could match or surpass the performance achieved by the Conditional Random Fields (CRF) for this particular task. While these alternative models achieved F1 scores of 25, 49, and 18 respectively, with Naive Bayes being the most successful among them, they still fell short compared to the CRF.

Given the superior performance of the CRF, we decided to focus our efforts on optimizing its hyperparameters rather than switching to a different algorithm. We explored various hyperparameters, including different algorithms such as L-BFGS with L1/L2 regularization, SGD with L2-regularization, Averaged Perceptron, Passive Aggressive, and Adaptive Regularization of Weights (AROW), as documented in the CRFSuite documentation. However, none of these alternatives provided an improvement over the initial SGD with L2-regularization.

Additionally, we fine-tuned the regularization parameter in an attempt to enhance the accuracy of our CRF model. Although we made a slight adjustment by reducing the regularization parameter from 0.1 to 0.08, we did not observe a significant difference in performance.

Changing the minimum frequency of a feature did not improve the performance of the model.

```
# Use L2-regularized SGD and 1st-order dyad features.
trainer.select('l2sgd', 'crf1d')

# This demonstrates how to list parameters and obtain their values.
trainer.set('feature.minfreq', 1) # minimum frequency of a feature to consider it
trainer.set('c2', 0.08)           # coefficient for L2 regularization previously to .1

print("Training with following parameters: ")
for name in trainer.params():
    print (name, trainer.get(name), trainer.help(name), file=sys.stderr)

# Start training and dump model to modelfile
trainer.train(modelfile, -1)
```

3 Feature Extractor

We experimented with various combinations and parts of words to enhance the classification of entities. Initially, we applied a simple transformation to tokens, converting them to lowercase to avoid inconsistencies at the beginning of sentences, resulting in a modest improvement from 55.5% to 56.7% in F1 Score.

Subsequently, we extracted the last character of each word to discern whether the token represented a plural form, which notably boosted the F1 Score to 65.3%. Following this, we considered the length of the word, hypothesizing that longer words might correlate with specific categories, leading to a further improvement to 67.8% in F1 Score.

Further refinement involved extracting the first four characters of tokens, particularly focusing on entities starting with "anti-", which contributed to a substantial enhancement in performance, yielding an F1 Score of 71.7%. Additionally, we identified the number of capital characters present in tokens, which often indicated the presence of abbreviations and improved the model's performance to 74.1% in F1 Score.

Although we experimented with features such as the presence of numbers, dashes, and parentheses, none of them resulted in a notable improvement in the F1 Score. Therefore, adhering to the principle of parsimony, we retained only the features that demonstrated significant enhancements in the model's performance.

3.1 Code

```
## --------- Feature extractor -----------
## -- Extract features for each token in given sentence
def _contains_numbers(word):
    return str(bool(re.search(r'\d', word)))

def _contains_dashes(word):
    return str(bool(re.search(r'-', word)))

def _count_uppercase_characters(word):
    return str(sum(1 for char in word if char.isupper()))

def _first_character_is_upper(word):
    return str(word[0].isupper())

def _has_parentheses(word):
    return str(any(char in '()' for char in word))

def extract_features(tokens) :
   # for each token, generate list of features and add it to the result
   result = []
   for k in range(0,len(tokens)):
      tokenFeatures = [];
      t = tokens[k][0]

      tokenFeatures.append("lower_form="+str(t).lower())
      tokenFeatures.append("suf3="+t[-3:])
      tokenFeatures.append("suf1="+t[-1:])
      tokenFeatures.append("token_len="+(str(len(t))))
      tokenFeatures.append("prefix4="+t[:4])
      # tokenFeatures.append("numbers="+_contains_numbers(t))
      # tokenFeatures.append("dashes="+_contains_dashes(t))
      # tokenFeatures.append("parenthesis="+_has_parentheses(t))
      tokenFeatures.append("n_capital="+_count_uppercase_characters(t))

      if k>0 :
         tPrev = tokens[k-1][0]
         tokenFeatures.append("formPrev="+tPrev)
         tokenFeatures.append("suf3Prev="+tPrev[-3:])
      else :
         tokenFeatures.append("BoS")

      if k<len(tokens)-1 :
         tNext = tokens[k+1][0]
         tokenFeatures.append("formNext="+tNext)
         tokenFeatures.append("suf3Next="+tNext[-3:])
```

```
        else:
            tokenFeatures.append("EoS")

        result.append(tokenFeatures)

    return result
```

4 Experiments

These are the accumulated step-by-step improvements made by each feature.

Transformation Applied	F1 Score (accumulated) (%)
Lowercase	56.7
Last character of each word	65.3
Length of the word	67.8
First four characters of tokens	71.7
Number of capital characters	74.1

These are the specific metrics with all the features extracted.

Category	tp	fp	fn	pred	exp	P	R	F1
brand	301	12	73	313	374	96.2%	80.5%	87.6%
drug	1712	127	194	1839	1906	93.1%	89.8%	91.4%
drug_n	10	8	35	18	45	55.6%	22.2%	31.7%
group	566	71	121	637	687	88.9%	82.4%	85.5%
M.avg	-	-	-	-	-	83.4%	68.7%	74.1%
m.avg	2589	218	423	2807	3012	92.2%	86.0%	89.0%
m.avg (no class)	2643	164	369	2807	3012	94.2%	87.7%	90.8%

5 Conclusions

In this project, we delved into the Named Entity Recognition and Classification (NERC) task with a focus on identifying and categorizing drug names within unstructured text data. Our approach centered on implementing a Sequence Tagging methodology using the B-I-O schema and leveraging a Conditional Random Fields (CRF) model, which demonstrated superior performance over other machine learning algorithms tested.

Through experimentation with feature extraction and model optimization, we achieved significant improvements in the model's accuracy. Key features such as the number of capital characters in tokens and the first four characters of words were instrumental in enhancing the model's performance. Additionally, we meticulously tested various hyperparameters for the CRF model and fine-tuned the regularization parameter, aiming to optimize accuracy further.

The final CRF model achieved an F1 Score of 74.1%, showcasing its ability to accurately classify entities into different categories when tested on a relevant dataset. These results underscore the critical role of feature extraction in NERC tasks and affirm the efficacy of the CRF algorithm for entity recognition within unstructured text data.

However, it's important to note a significant area for improvement highlighted by our experiments: the classification of 'drug_n' entities. The model struggled with this category, primarily due to the limited instances of 'drug_n' in the training data and its similarity to the 'drug' category. This challenge underscores the need for more nuanced feature extraction and potentially more sophisticated modeling techniques to distinguish between closely related categories more effectively.

Future directions for this project could include exploring additional features that could help differentiate between such closely related categories, optimizing hyperparameters even further, and investigating the application of deep learning techniques to enhance the model's performance. By addressing the specific challenge of 'drug_n' classification and continuing to refine our approach, we can work towards developing a more accurate and robust NERC system.

Overall, this project serves as a valuable exploration into the application of machine learning techniques for entity recognition and classification in unstructured text data, highlighting both the achievements and the areas for improvement in the pursuit of more sophisticated NERC systems.