Innopolis University
Summer 2024 Trimester
Natural Language Processing Course

Assignment #3 Solution Report - Russian Nested Named Entities Recognition

Student: Hamada Salhab
Group: BS-21-AI-01
April 2024

Email: h.salhab@innopolis.university
Codalab username: hamadasalhab
Github repository: https://github.com/HamadaSalhab/nlp-assignment-3

1. Introduction

The task of extracting nested named entities from Russian texts presents unique challenges, particularly due to the depth of nesting and the variety of entity classes involved. This report outlines two distinct approaches to tackle this problem, utilizing different methodologies to optimize entity recognition performance on the NEREL dataset.

2. Solution Overview

2.1 Solution 1: Frequency-Based Tagging

The solution can be found in the *solution-1* directory in the Github repository. The submission data generated using this approach can also be found in the same directory under names “test.jsonl” and “test.zip”.

2.1.1 Approach

This solution employs a simplistic yet intuitive approach where a dictionary is created to store the most frequent named entity recognition (NER) tags for each token based on the training dataset. This dictionary serves as a lookup table during the prediction phase on the test dataset.

2.1.2 Methodology

- Data Reading and Preprocessing: Tokens are extracted and their frequencies are recorded from the training set.
- Dictionary Construction: A dictionary is built where each token is associated with its most frequently occurring tag.
- Prediction: The test data sentences are tokenized and matched against the dictionary to predict the NER tags based on previously observed frequencies.

2.1.3 Code Snippets

- The dictionary is constructed using token counts and the associated NER tags are stored as the most frequent tag observed with each token.
- Predictions on the test set are made by tokenizing the sentences and retrieving the tag from the dictionary for each token.

2.1.4 Results

- F1-Score: 0.33
- This method, while straightforward, showed limitations in handling the context and complexity of nested entities.

2.2 Solution 2: CRF with Feature Engineering [Reference]

The solution can be found in the *solution-2* directory in the Github repository. The submission data generated using this approach can also be found in the same directory under names “test.jsonl” and “test.zip”.

2.2.1 Approach

This method utilizes Conditional Random Fields (CRF), enhanced with a comprehensive set of linguistic features extracted from the text using the Russian spaCy model.

2.2.2 Methodology

● Feature Extraction: Features such as the lowercased token, whether the token is uppercase, titlecase, part-of-speech tags, and surrounding context (previous and next token features) are extracted.

● Model Training and Evaluation: A CRF model is trained with these features. The model is initially evaluated on a split of the training data, then retrained on the entire dataset for final predictions.

● Prediction: Features are extracted from the test set and fed into the CRF model to generate predictions.

2.2.3 CRF Parameters

In the CRF model configuration, the following parameters were utilized:

- Algorithm: 'l2sgd' - This parameter sets the training algorithm to Stochastic Gradient Descent with L2 regularization term. It's an effective choice for CRF, as it supports large-margin structured learning, which is crucial for handling the high dimensionality in NER tasks.
- max_iterations: 200.
- all_possible_transitions: True - Allows the model to consider all possible transitions between labels in adjacent tokens, even those not observed in the training data..

2.2.4 Results

● F1-Score: 0.41

● The CRF model outperformed the frequency-based approach, demonstrating better handling of contextual dependencies and nested entity structures.

3. Conclusion and Recommendations

The comparative effectiveness of the CRF model highlights the importance of contextual and morphological features in NER tasks, especially for complex scenarios like nested entity recognition. Future work could explore hybrid models combining deep learning techniques with CRF for potentially better performance.