NLP Assingment 3

Timolai Andrievich
t.andrievich@innopolis.university
CodaLab nickname: tandrievich

April 2024

GitHub repository link: https://github.com/timolai-andrievich/nlp-assignments (see "assignment3" directory).

1 Introduction

Named entity recognition (NER) is an information extraction task. The goal of this assignment is to create models for the NER task, and evaluate them using NEREL-based Russian dataset containing 29 classes.

2 Approaches

2.1 Naive approach

The most obvious approach for the NER task is to split the text into words through some approach and count the labels for each word in the training set. During inference, the most common label is predicted for each word. Being the simplest approach listed, it will be used as a baseline.

2.2 N-grams

N-grams are often the next step after the naive approach in NLP tasks, as they allow for incorporation of a limited context, while still being easy to implement and having a good performance.

2.3 Fine-tuning a BERT model

BERT [1] is a model introduced in 2019 by Devlin et al., but the name since became the name of the family of tranformer models, the main characteristics of which is bidirectionality and pre-training on masked token prediction and next sentence prediction. This encourages the model to produce meaningful token embeddings, which then can be used in downstream tasks, such as NER. Furthermore, using modern tooling, finetuning such a model is trivial.

3 Method

For the first two approaches, the text is split into tokens using regular expression `\S+`, which effectively splits the text by whitespace symbols. For each approach, two predictions are joined if they both predict the same label, and are separated by one character. This is done to match the dataset format, in which multiple words can belong to the same entity.

3.1 Naive approach

For each token in the text, the labels are counted. The label is counted if the word intersects with the entity with the corresponding label. If one word overlaps with multiple entities, label for each entity is counted. If the word doesn't overlap with any of the entities, the no-entity label is counted instead. During inference, the most common label for each word is chosen. If the word was not seen during training, the no-entity label is predicted.

3.2 N-grams

For each word in the text, two N-grams are formed: the one with preceding context, and the one with succeding context, each containing the target word. The text is padded beforehand, so the words in the start and the end of the text are counted as well. Then the labels are counted for both n-grams. Labels are counted separately for preceding and succeding contexts, and individual words are labeled the same as in the naive approach. During inference, the label counts for preceding and succeding counts are summed, and the most common label from the joint counts is chosen.

3.3 BERT

During the training, the tokens are labeled the same way the words are labeled in the naive approach: if the token intersects an entity, it is labeled using the label of the entity, otherwise it is labeled with the no-entity label. It is assumed that no entities overlap, and each token can overlap with one entity at most. Then, the model is trained on the resulting dataset. As the pretrained model, two models were used: LaBSE [2], and the multilingual version of BERT [1]. During inference, each word was labeled with the entity of the first token in the word.

4 Evaluation

The models were evaluated using character-wise F1 score, which is calculated as follows: each character is labeled by the label of the token it belongs to. If a symbol does not belong to any token, it is labeled as no-entity. This operation is performed on both prediction and the target data, and macro-averaged F1 score is calculated on the resulting labeled strings.

Model	Symbol-wise F1 score	F1 score
Naive	0.7260	0.2056
Bigram	0.6809	0.0868
Trigram	0.5330	0.0241
BERT	0.8562	0.5479
LaBSE	**0.8742**	**0.5786**

Table 1: Evaluation results

As seen in Table 1, n-gram models perform worse than the baseline naive approach. This could be due to a multitude of reasons, most likely being the small size of the training set, leading to most of bigrams and trigrams not being seen during training.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[2] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic bert sentence embedding, 2022.