National Research University Higher School of Economics

Faculty of Computer Science

Programme ‘Master of Data Science’

MASTER’S THESIS

Machine reading comprehension methods for named entity recognition

Student Lyubov N. Sharaborina

Supervisor: PhD, Ekaterina L. Artemova

Moscow, 2022

**Contents**

[1. Introduction 3](#_Toc104650221)

[2. Foundation 5](#_Toc104650222)

[2.1. Natural language processing 5](#_Toc104650223)

[2.2. Machine reading comprehension 7](#_Toc104650224)

[2.3. Named entity recognition 8](#_Toc104650225)

[2.4. Pretrained models and transfer learning 11](#_Toc104650226)

[2.5. BERT 12](#_Toc104650227)

[2.6. SBERT 13](#_Toc104650228)

[2.7. MRC-NER framework 14](#_Toc104650229)

[2.8. NEREL dataset 16](#_Toc104650230)

[3. Experiment 20](#_Toc104650231)

[3.1. Query generation 20](#_Toc104650232)

[3.2. Data preprocess 22](#_Toc104650233)

[3.3. Training and evaluating BERT MRC-NER 23](#_Toc104650234)

[3.4. Analyzing results 24](#_Toc104650235)

[4. Conclusion 30](#_Toc104650236)

[References 31](#_Toc104650237)

[Appendix 32](#_Toc104650238)

# Introduction

Currently, there is an exponential growth in the volume of information in unstructured and raw form, as a rule, the World Wide Web consists of just such data. Extracting useful information from unstructured data and sorting, classifying, analyzing it is a difficult task. And it is obvious that due to the amount of information for such tasks, there is a catastrophic lack of human resources. Thus, the development of algorithms and methods capable of automatically analyzing information, structuring and presenting it in a form understandable to a person, becomes an urgent task.

For a machine, the knowledge contained in unstructured documents can be accessed and brought into a structured form thanks to the development of artificial intelligence (AI) methods. As recent achievements in AI have shown, such methods can quite successfully solve a number of problems in the field of natural language processing on their own without human intervention. Natural language processing (NLP) is one of the most important areas of research in computer science, dealing with the analysis of documents based on a variety of theories and methods. An important part of natural language processing is the extraction of useful information: objects, relationships between objects. This process can be described as understanding the text without human intervention. Now many companies are actively using interactive systems, such as chat bots, assistant robots, to increase the throughput of call centers and reduce their maintenance costs. Artificial intelligence in the field of natural language processing can help technical support operators find the correct answer faster or communicate with people at the initial stage of a dialogue. Also, on news sites, it is important to keep all articles in a structured form, artificial intelligence can itself classify a document into the desired heading, or offer a number of suitable headings for a person to choose from. In social networks, online cinemas and other sources containing entertainment content, the user constantly needs to recommend new content based on previous viewing, and sometimes what he himself does not know about.

Artificial intelligence can become an indispensable assistant to a person in sorting letters in the mail, executing human commands, finding out the weather forecast, information about traffic jams, and managing one or another smart home device. Voice assistants are now being introduced everywhere, and this also lies in the field of natural language processing. The main task of AI is to understand natural language as well as a person, use the accumulated knowledge of mankind, analyze input data and constantly learn. Therefore, all this makes this task relevant and promising.

Recently there has been a trend of converting NLP problems into machine reading comprehension (MRC) tasks. The concept of MRC is similar to how a person understands a text. The most common way to check is to ask a person to answer questions on a read piece of text. Similarly, the computer's ability to understand language is evaluated. For example, McCann et al. (2018) formalized summarizing tasks to a question answering task ("What is this text about?"). [Li et al., 2019]

In comparison with sequence labeling, application of MRC method has two significant advantages:

1. The MRC formulation of the NER task can help to handle overlapping entities.
2. In MRC formulation, the query encodes significant prior information about the entity category, so that the model has the potential to disambiguate similar tagging classes. [Li et al., 2019]

In this work, a Unified MRC Framework [ShannonAI., 2020], based on a pre-trained BERT model, was applied. This architecture showed good results on the English ACE 2005 dataset using the pretrained model in English bert-large-uncased [Devlin et al., 2018], so we decided to use it when training the model based on the Russian NEREL dataset using the pretrained sbert-large-nlu-ru model [Antukhov, 2020].

The goal of this work is to apply machine reading comprehension (MRC) methods based on neural networks for training on the Russian NEREL dataset for the recognition of the named entities.

# Foundation

## Natural language processing

Natural Language Processing (NLP) can be framed as a branch of Computer Science and AI that focuses on how computers understand, parse, manipulate, and potentially generate natural languages ​​[Zhu, 2021]. In other words, NLP is an intersection of machine learning and computational linguistics that studies the methods of analysis and synthesis of natural languages ​​[Collobert et al., 2011]. Today, NLP is used in many areas such as speech recognition, document annotation and summarization, machine translation, named entity recognition, chatbots, autocomplete, voice assistants, spam detection etc. NLP techniques help to analyze human language patterns and structures in order to develop computer models for language understanding and text creation. [Collobert et al., 2011]

NLP solves a large set of problems. Among these tasks are the following:

1. Text recognition, speech recognition, speech synthesis;
2. Morphological analysis, canonization;
3. Part of speech (POS) tagging, named entity recognition (NER), word highlighting;
4. Syntax parsing, tokenization of sentences;
5. Relation extraction, language detection, sentiment analysis;
6. Annotation of the document, translation, analysis of topics;
7. Deduplication, information retrieval. [Zhu, 2021]

Machine learning models take digital data as input. When working with text, the first thing to do is go to its digital representation. The task of natural language models is to represent text in vector form. This process is called "embedding", which means that the meaning of the words is digitized by the model and written as an ordered set of numeric values ​​(i.e. a vector).

The first models, such as Word2Vec, "understood" the meaning of the text only at the level of individual words, without context. Recent advances in computational linguistics have made it possible to move towards efficient vector representation of entire sentences and paragraphs of text [Collobert et al., 2011].

The location of phrases in the vector space is determined by the parameters of the language model used. Models are trained in such a way that the vectors calculated by them preserve the semantic relationships between phrases - sentences similar in meaning are encoded into vectors close in metrics. Therefore, based on them, it is convenient to apply instance-based learning techniques, for example, the K-NN nearest neighbors method [Collobert et al., 2011].

At the moment, the most effective way to build natural language models is to train deep neural networks based on the "transformer" architecture: BERT, RoBERTa, GPT-3 [Zhu, 2021].

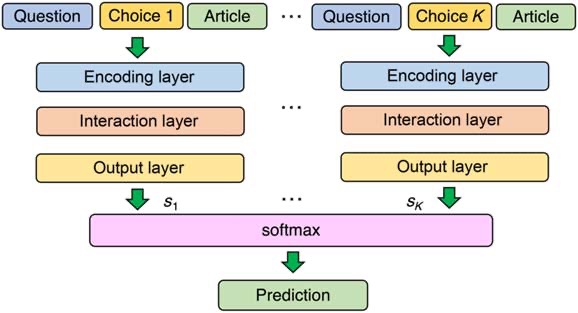
These models are universal and are able to extract features from the text that are useful for solving many problems of text analysis. For this reason, they are sometimes called Natural Language Understanding Models or NLUs [Zhu, 2021].

## Machine reading comprehension

Machine reading comprehension (MRC) is one of the most important tasks in language understanding [Zhu, 2021]. MRC task can be formulated as a supervised learning task: the model receives a text and a related question, and it should produce an answer in the required format [Zhu, 2021].

MRC models extract response spans from the text. And it can be considered as two multiclass classification task – to make a prediction of the start and end position of the response spans [Zhu, 2021].

Architecture of an MRC model consists of the encoding layer, the interaction layer, and the output layer. The encoding layer generates word embeddings, the interaction layer combines information from the text (article) and the question, and usually uses a cross-attention mechanism, the output layer generates answers.



[Zhu, 2021]

Since 2015, MRC algorithms have mostly been built on deep learning and deep neural networks. These approaches use their enormous model complexity to accurately characterize the semantic space and achieve higher response fidelity [Zhu, 2021]. Additionally, it should be mentioned that these models usually do not require hand-crafted features. Instead, a generalizable feature representation can be automatically extracted from the data. This significantly reduces dependence on expert knowledge and subsequent linguistic tools. The success of deep learning models is also closely related to the advent of large-scale datasets such as SQuAD, RACE, and MS MARCO [Zhu, 2021].

## Named entity recognition

“Named entities refer to proper nouns, which are generally divided into such categories as: names of person, location, organization, time and date etc.” [Zhu, 2021]. The task of NER is to highlight spans of entities in the text. NER is supposed to understand that the passage of a text "April 7, 2022" is a date, "Moscow" is a location, and "HSE" is an organization [Zhu, 2021].

There can be extracted two main tasks in NER problem:

* + - 1. to detect that some sequence of words is a named entity;
      2. to classify entity type (for example, name of a person, name of an organization, city, etc.)

Each stage has its own difficulties. There is no standard set of named entity classes. Generally, they try to extract the names of people and the names of places and organizations, but the compound of entity types usually depends on the specific task that need to be solved, or on the capabilities of the pre-trained system that is planned to be used. Additionally, the NER task includes extracting such entities as dates, money, “which do not intuitively look like named entities” [Smurov, 2019].

Many named entities can refer to different classes depend on a context: the word "Pushkin" can be a person, a city, a club name, etc. Understanding exactly which class a word belongs to in a particular context is a difficult task, which is now a lot of work.

There are following general NER methods:

1. Rule-based NER

Named entities can be extracted without machine learning algorithms. And in that case rule-based systems (in the simplest version – regular expressions) can be helpful. For example, if you need to highlight emails or numeric entities (dates, amounts of money, or phone numbers), regular expressions can help to build a rule, by which this information can be taken [Zhu, 2021].

Rules can be edited manually to recognize names of person, location, and organization: for example, the word after "Mrs." or "Mr." is more likely to be the name of a person [Zhu, 2021].

However, because of words’ ambiguity, such methods cannot perform good results. Therefore, it makes sense to use them only for limited domains and for simple and clearly separable entities.

1. Feature-based NER

Since NER is essentially a word classification task, it belongs to supervised machine learning when labeled data is available. In addition, NER is most commonly a sequence labeling problem since tags are predicted for a sequence of words. Therefore, the label of a word depends not only on the word itself, but also on neighboring words and labels. Feature-based NER extracts features for various aspects of a word and its context, as in the following examples:

1. whether it is a noun;

2. whether the word is enclosed in quotation marks;

3. whether the word is the first or the last word of the sentence;

4. word’s prefix and suffix.

Once the characteristics of each word are obtained, NER can be performed using context-independent models (e.g., logistic regression) or context-dependent models (e.g., conditional random field) [Zhu, 2021].

The scheme of a sequence labeling task is to add some prefix to the entity label (for example, PER for persons or ORG for organizations), which indicates the position of the token in the entity span. In details:

“B – the first token in the entity span that consists of more than one word.

I – the words inside the span.

E – the last token of an entity that consists of more than one word.

S – single, the entity consists of one word.”[Smurov 2019, 2021]

These prefixes are added to each entity type. If the token does not belong to any entity, it is marked with a special label OUT or O.

This method is widely used, but it has several significant defects.

The main drawback is that the schema does not allow to deal with nested entities. For example, the entity “Lyceum named after P.L. Kapitsa” is an organization. But Piotr Kapitsa is a person. Using the markup method described above, it will not be possible to convey both of these facts. [Smurov 2019, 2021].

There is a standard data format that is convenient for storing markup for a NER task (as well as for many other NLP tasks). This format is called CoNLL-U.

The main idea of ​​the format is as follows: to store data in the form of a table, where one row corresponds to one token, and columns correspond to a specific type of token features (including the word itself, the word form, is a feature) [Smurov 2019, 2021].

1. NER based on Deep learning

Neural networks take vectors of words and use a multiclass classifier layer to output likelihood scores of each word to be a named entity of a certain type. For example, there are several scores: 0.47, 1.16, 16.13 and 0.89 for different entity types. These scores are then normalized using softmax and compared to ground-truth labels [Zhu, 2021].

The advantage of this method is that all model parameters can be optimized simultaneously to improve NER performance in an end-to-end manner. Deep learning can flexibly adjust the size of the network to improve performance when working with large-scale datasets. As a result, the best NER models are currently based on deep learning [Zhu, 2021].

## Pretrained models and transfer learning

Pre-trained models were first applied in the computer vision. In 2012, AlexNet deep learning model won first place in ImageNet – image recognition competition, well ahead of other methods. AlexNet has been widely used in many computer vision tasks. Instead of retraining the AlexNet network architecture from scratch, many new models reuse its parameters and continue to fine-tune the network for the target tasks. Experimental results show that the use of pretrained AlexNet significantly improves the accuracy of solving the target problem and significantly reduces the training time for a new model. The reason is that the pretrained model already has a strong ability to understand images after being exposed to massive data in the original task. This method of transferring a model trained on one task to other related tasks is called transfer learning. [Zhu, 2021]

Transfer learning is similar to how people use their prior knowledge and skills to solve several similar tasks. For example, it is much faster for an interpreter to learn a new language than it is for a person with no experience in foreign languages to complete the same task, because an interpreter can transfer and adapt their learning skills to a new language. In machine learning, transfer learning first trains a model A on the original task(s) N and then optimizes it on the target task K to get the final model A\_new. In order to ensure the efficiency of model transfer, tasks N and K must be related and have some similarity. [Zhu, 2021]

One of the main reasons for the popularity of transfer learning is that there is often not enough data for the target task. For example, in many MRC tasks, questions and answers must be manually generated and edited, which is time consuming and labor intensive. If a large model is trained from scratch on a small dataset, overfitting can occur, reducing the generalizability of the model. Transfer learning alleviates this problem by pretraining the model on large-scale data of one or more related tasks before adapting to the target task. [Zhu, 2021]

Transfer learning is gaining popularity as more and more pretrained models are in the public domain, so the code and trained parameters of these models can be easily accessed. It saves a significant amount of time and computational resources by directly starting from a checkpoint of a pretrained model to further tune it to the target task. [Zhu, 2021]

## BERT

In October 2018, Google introduced a pretrained Bidirectional Encoder Representations from Transformers (BERT). BERT ranked first in 11 NLP tasks and outperformed human-level performance in SQuAD v1.0 and in adversarial generation (SWAG) common sense reasoning datasets. Once the code and pretrained model were released, BERT was immediately adopted in various NLP models and achieved remarkable results. For instance, in the CoQA dataset, all of the top 10 models are based on BERT; in the SQuAD v2.0 dataset, all top 20 models are based on BERT. [Zhu, 2021]

BERT has a multilayer Transformer structure. Its input text is tokenized with WordPiece, a tokenization method similar to Byte Pair Encoding (BPE), and its outputs are BERT context embeddings that encode the word's meaning and context information. In pretraining, the BERT model solves two tasks: a masked language model (MLM) and next sentence prediction (NSP). Both tasks fall under the category of self-supervised learning, which requires only a text corpus and doesn’t need any manual labels. [Devlin et al., 2018]

BERT proposes to mask the input words. It randomly selects 15% of the input words to be replaced by the special symbol [MASK]. It then uses the multilayer Transformer model to predict the masked words. As these words have been completely masked in input, it is valid to use the input to predict them. As a result, BERT is a bidirectional language model. [Devlin et al., 2018]

BERT is pretrained on public corpus including BooksCorpus (800 M words) and English Wikipedia (2500 M words). There are two versions of BERT models available online:

• BERT BASE: 12-layer Transformers, input and output dimension of

768, 12 attention heads with 110 M parameters;

• BERT LARGE: 24-layer Transformers, input and output dimension of

1024, 24 attention heads with 340 M parameters. [Devlin et al., 2018]

# SBERT

SBERT is a NLU model by SberDevices. SBERT is based on BERT and datasets SNLI, MNLI (for training) and STS SICK (for validation) [Antukhov, 2020].

The architecture of the model is a Siamese neural network with three inputs for the “anchor” – “positive” – “negative” triplet. A BERT module is applied to each of the inputs, which will play the role of NLU in this experiment. The module contains a WordPiece tokenizer for converting input strings into a BERT format (input\_ids, input\_mask, token\_type\_ids), as well as a trainable BERT model for text vectorization. We apply the masked-mean-pooling operation to the output of the last encoder layer of the model to get a single vector for the proposal. The triplet vectorized in this way is then used to calculate the softmax loss and train the model. The model was trained on 16 GPUs [Antukhov, 2020].

Model SBERT has a following architecture:

Diagram

Description automatically generated

[Antukhov, 2020]

## MRC-NER framework

In 2019, Xiaoya Li et al. presented Unified MRC Framework for Named Entity Recognition [ShannonAI., 2020]. The framework is capable of handling flat and nested named entities.

Li et al. propose to formulate the NER task as a machine reading comprehension (MRC) task. They apply the MRC method, instead of treating NER as a sequence labeling problem. [Li et al., 2019]

And extracting named entities is formalized as extracting answer spans to the question. For example, for COUNTRY entity label it would look the following way:

Text: “Yakutia is the coldest region of Russia.”

Question: “which country is mentioned in the text”

And the task is to extract answer spans to this question: (6:7, “Russia”)

The MRC formulation of the NER task can help to handle overlapping entities. And the second advantage is that in MRC formulation, the query encodes significant prior information about the entity category, so that the model has the potential to disambiguate similar tagging classes. [Li et al., 2019]

The tagging-style annotated NER dataset is transformed to a set of (question, answer, context) triples. The backbone of the framework is BERT (Devlin et al., 2018).

The question and the text are concatenated, forming the combined string with [CLS] and [SEP] special tokens to denote question and text parts. Then BERT receives the combined string and outputs a context representation matrix , where is the vector dimension of the last layer of BERT, and the query representations can be dropped. [Li et al., 2019]

Span selection strategy is to have two binary classifiers, one to predict whether each token is the start index or not, the other to predict whether each token is the end index or not. This strategy allows for outputting multiple start indexes and multiple end indexes for a given context and a specific query, and thus has the potentials to extract all related entities. [Li et al., 2019]

Given the representation matrix E output from BERT, the model first predicts the probability of each token being a start index as follows:

Pstart = softmaxeach row(E · Tstart) (1)

is the weights to learn.

Each row of Pstart presents the probability distribution of each index being the start position of an entity given the query.

The end index prediction procedure is exactly the same, except that we have another matrix Tend to obtain the probability matrix Pend . [Li et al., 2019]

In the context X, there could be multiple entities of the same category. This means that multiple start indexes could be predicted from the start-index prediction model and multiple end indexes predicted from the end-index prediction model.

By applying argmax to each row of Pstart and Pend, there will be predicted indexes that might be the starting or ending positions. Given any start index and end index, a binary classification model is trained to predict the probability that they should be matched, given as follows:

Pistart,jend = sigmoid(m· concat(Eistart , Ejend ))

where m ∈ 1×2d is the weights to learn. [Li et al., 2019]

At test time, start and end indexes are first separately selected. Then the index matching model is used to align the extracted start indexes with end indexes, leading to the final extracted answers.

For nested NER, experiments are conducted on ACE 2004, ACE 2005, GENIA and KBP2017 datasets.

For flat NER, experiments are conducted on English datasets CoNLL2003 and OntoNotes 5.0, and Chinese datasets OntoNotes 4.0 and MSRA.

The MRC method obtains SOTA results on both nested and flat NER datasets. [Li et al., 2019]

## NEREL dataset

In this work we use the NEREL dataset proposed by Natalia Loukachevitch, Ekaterina Artemova, Tatiana Batura et al. NEREL is a new Russian dataset with annotated named entities and relations [Loukachevitch et al., 2021]. It is the largest dataset for the Russian language annotated with named entities and relations. NEREL features 29 entity and 49 relation types. The dataset contains 56.000 entities and 39.000 relations annotated in more that 900 Russian Wikinews documents. [Loukachevitch et al., 2021]

29 entity types of the NEREL dataset are following:

1. Basic entity types: PERSON, PROFESSION, ORGANIZATION, EVENT, and LOCATION (plus FACILITY, COUNTRY, STATE OR PROVINCE, CITY, DISTRICT).
2. Temporal and numerical entities: NUMBER, ORDINAL, DATE, TIME, PERCENT, MONEY, AGE.
3. Physical object group of entities: WORK OF ART, PRODUCT, and AWARD entities.
4. Nationalities, religious, or political groups: NATIONALITY, IDEOLOGY, RELIGION, LANGUAGE.
5. Legal entities: LAW, CRIME, PENALTY.
6. FAMILY entity, which is used to describe relations between families and their members, and DISEASE entity. [Loukachevitch et al., 2021]

Entities in the NEREL dataset can have a quite complicated nested structure, in particular, PROFESSION spans include names of corresponding organizations, ORGANIZATION and AWARD spans can include a name of a person. For example: “MSU named after M.V. Lomonosov” [Loukachevitch et al., 2021].

The dataset is divided into three parts: train set, dev set and test set. Each set consists of pairs: text file and file with annotation. Each text contains 10-20 sentences.

There are three bars which show the frequency of entity types in sets:

(Please, procedure to the next page)

1. For train set Chart, histogram

   Description automatically generated

Chart, histogram

Description automatically generated

1. For test set
2. Chart, histogram

   Description automatically generatedFor dev set

The largest group of entities in the dataset is: PERSON, PROFESSION, EVENT, ORGANIZATION, DATE, for all three sets. And the smallest group is: FAMILY, LANGUAGE, PERCENT, RELIGION, DISTRICT.

Additionally, there are bars which shows the mean entity length (for every entity type) in sets (by symbols):

1. For train set

Chart, bar chart, histogram

Description automatically generated

1. For test set

Chart, bar chart, histogram

Description automatically generated

1. For dev set

Chart, bar chart, histogram

Description automatically generated

As we can see, the longest mean entity length in train and test sets is LAW entity, in dev set – CRIME entity. It can be explained by the need to describe a situation of a crime and by the specifics of legal vocabulary.

# Experiment

## Query generation

To transform NEREL dataset into MRS format (context, question, answer) we need to create a list of queries for all entities which are presented in the dataset. These queries will play the role of questions in our task. The question/query generation procedure is important because queries encode prior knowledge about labels and can have a significant influence on the final results. [Li et al., 2020]

There are different ways to construct queries [Li et al., 2020]:

1. To use position index of labels: one, two, three
2. To formulate a keyword: a query is formulated as a keyword which describes the entity (for example, LOC is “location”).
3. Rule-based template filling, for example, LOC is “location which is mentioned in the text”.
4. To use dictionary: a query is constructed using a dictionary definition. The query for LOC is ” location or place are used to denote a region (point, line, or area) on Earth's surface or elsewhere.”
5. To use synonyms: for example, the query for LOC is “place”.
6. Keyword and synonyms.
7. Annotation guideline notes. For example, the query for tag LOC is ”find in the text geographical location, country of city”.

Queries for this task were formulated based on the last formula, namely, annotation guideline notes. Our queries start with the guideline ‘find in the text’ as it was proposed by Li et al. (Li et al., 2020) for ACE2005 dataset, we use this recommendation because ACE2005 considers nested NER and performs very good results in comparison with flat NER [Li et al., 2020].

During the process of creating a description of a named entity, we relied on data from the corresponding Wikipedia articles and explanatory dictionaries.

Examples of queries:

PERSON: "Найди в тексте обозначение человека, человеческого существа, имени или фамилии",

PROFESSION: "Найти в тексте обозначение рода трудовой деятельности человека, обозначение труда или занятия, за которое человек получает оплату",

ORGANIZATION: "Найди в тексте обозначение производственного образования, сформированного из людей, деятельность которых координируется руководством для достижения общей цели"

DISEASE: "Найди в тексте обозначение расстройства здоровья, нарушающего деятельность организма или его отдельных органов",

LOCATION: "Найди в тексте обозначение места или местоположения какого-либо объекта или предмета",

IDEOLOGY: "Найди в тексте обозначение системы взглядов и идей, характеризующих какую-либо социальную группу, класс, политическую партию или сообщество".

## Data preprocess

On this stage we were going to transform tagging-style annotations to a set of MRC-style (Context, Query, Answer) triples.

We had train, test and dev sets of data, and all texts were provided with annotation, which contained entity labels and start/end positions in text (by symbols).

We had already formulated a list of queries on the previous step and we needed to combine this list of queries with our data and adapt the dataset to following template:

{Context: “text ”, query: “text ”, start\_position: (number), end\_position: (number)}

We cleaned the text from special symbols (\n, \t etc.), hieroglyphs, divided all texts into sentences. After that we tokenized every sentence and created a map between text ‘s indexes and sentences’ indexes to modify position spans from annotation.

We also needed to change the values of the entity’s start and end character to the values of the start and end word.

Next step was to combine sentences and query by entity label. We added a new start position and end position, and provided every sentence with an id number.

This procedure was performed for every text from the corresponding set and resulting sentences were appended to a file of JSON format.

There is an example of a scheme of a resulting dataset’s element:

"context": context,

"query": query,

"start\_position": [int(x.split(",")[0]) for x in positions],

"end\_position": [int(x.split(",")[1]) for x in positions],

"qas\_id": f"{origin\_count}.{tag\_idx}"

## Training and evaluating BERT MRC-NER

As it was previously mentioned, first we trained the model on the ACE 2005 dataset to examinate the perfofmance of the framework. ACE 2005 is a dataset for English, Arabic, and Chinese proposed by the Linguistic Data Consortium (LDC), it and contains 1,800 texts and 7 entity types.

We trained 20 epochs and saved top 3 of them. After that we evaluate checkpoints.

Li et al. used F1 metric, which is the harmonic mean of the precision and recall, to measure the model performance. And after training step we got the following results for ACE 2005:

f1: 0.8508

precision: 0.8353

recall: 0.8670

And to compare we present the result proposed by Li et al. for ACE 2005 for BERT-MRC:

f1: 0.8688

precision: 0.8716

recall: 0.8659

We can say that the result we received is close to the result of Li et al. experiments. Next step is to train MRC-NER on modified NEREL dataset. We trained 20 epochs on 2 GPUs and saved top 3 of them. The model evaluation result is following:

f1: 0.7721

precision: 0.8234

recall: 0.7269

## Analyzing results

We had allocated a small part of the dataset to conduct an experiment to make inference, evaluate and analyze the model predictions.

The test sample contains 1126 sentences. Some sentences are empty – they do not contain any entity.

We received the result in a text format, and we needed to preprocess the result first before analyzing it.

There are several steps we made:

1. Split the text, so that we will deal with a list of strings containing the following information: context (sentence), answer from the dataset, prediction of the model.
2. Iterate through list and clean its elements: remove additional (duplicate) information and special symbols (%, \n)
3. Transform list to a dictionary, where keys are entity labels, values are triplets (sentence, answer from the dataset, prediction of the model)
4. Next step is to transform values of every dictionary’s key to Pandas Dataframe with following columns: Context, True, Predict.
5. We also iterate through every row to compute the length of the predicted sting and its similarity score to true value (from the NEREL dataset).

We computed the total number of predictions, which is equal to 3645.

This bar shows the distribution of the number of predictions across all types of named entities:

Chart, histogram

Description automatically generated

As we can see in the previous bar, the largest top 5 predicted group is: PERSON, PROFESSION, ORGANISATION, EVENT, DATE.

This can be explained by the fact that these types of entities are the largest group of entities presented in the NEREL dataset.

For these entity types we made a plot of dependency between the length of predicted entity and similarity to corresponding entity value from the NEREL dataset.

Chart, bar chart

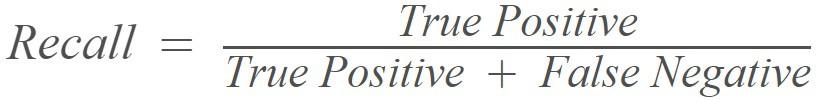
Description automatically generated

The best performance is shown for short entities (from 1 up to 3 words). This length is most common for all entity types. And we have a hypothesis that the types with the highest mean length (LAW, CRIME, AWARD) have lower performance than other types. It will be considered below.

We also computed F1 scores for predictions of every entity type.

As we remember F1 is a metric is based on the concepts of precision and recall, where:





In our case, we consider True Positive, True Negative, False Positive and False Negative values through the prism of the dataframe of each named entity. Dataframe’s columns are: Context, True, Predict. And we looked at the last two columns. Values of the column ‘True’ are the ‘Golden Standard’, and it could be empty, if there was no entity in the sentence. And values of the column ‘Predict’ are our predictions, and it could be empty, if there was not a predicted entity in the sentence. David Batista [Batista, 2018] suggests to modify TP, TN, FP and FN and to define number of Correct, Incorrect, Partial, Missing, Spurius values, where:

1. Correct (COR) – cases, where both “True” and “Predict” are the same;
2. Incorrect (INC) – “Predict” and “True” don’t match;
3. Partial (PAR) – “Predict” and “True” are somewhat “similar” but not the same;
4. Missing (MIS) – “True” is not captured by a system;
5. Spurius (SPU) – system produces a “Predict” which doesn’t exist in “True”. [Batista, 2018]

And we will compute Precision and Recall by these formulae [Batista, 2018]:

F1 score will be computed by the formula:

The F1 scores for the largest entity types are:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Label** | **TP** | **possible** | **actual** | **precision** | **recall** | **f1** |
| PROFESSION | 591 | 850 | 806 | 0.7332 | 0.6953 | 0.7138 |
| PERSON | 825 | 955 | 993 | 0.8308 | 0.8639 | 0.8470 |
| DATE | 471 | 521 | 558 | 0.8441 | 0.9040 | 0.8730 |
| ORGANIZATION | 438 | 658 | 665 | 0.6586 | 0.6657 | 0.6621 |
| EVENT | 348 | 659 | 561 | 0.6203 | 0.5281 | 0.5705 |

The highest F1 scores have following entity types:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Label** | **TP** | **possible** | **actual** | **precision** | **recall** | **f1** |
| PERSON | 825 | 955 | 993 | 0.8308 | 0.8638 | 0.8470 |
| DATE | 471 | 521 | 558 | 0.8440 | 0.9040 | 0.8730 |
| AGE | 120 | 146 | 128 | 0.9375 | 0.8219 | 0.8759 |
| COUNTRY | 408 | 456 | 449 | 0.9086 | 0.8947 | 0.9016 |
| RELIGION | 19 | 23 | 19 | 1.0000 | 0.8260 | 0.9047 |

CRIME, AWARD and LAW entity types have F1 score 0.5517, 0.5952 and 0.5983 respectively. It is not the lowest F1 score, but it corresponds to the lower half of the list of scores.

The lowest F1 scores have the following entity types:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Label** | **TP** | **possible** | **actual** | **precision** | **recall** | **f1** |
| FAMILY | 1 | 13 | 4 | 0.2500 | 0.0769 | 0.1176 |
| PENALTY | 7 | 18 | 14 | 0.5000 | 0.3888 | 0.4375 |
| FACILITY | 29 | 63 | 64 | 0.4531 | 0.4603 | 0.4566 |
| LOCATION | 24 | 61 | 41 | 0.5853 | 0.3934 | 0.4705 |
| DISTRICT | 11 | 25 | 17 | 0.6470 | 0.4400 | 0.5238 |

FAMILY entity is the smallest entity type presented in the train set. And it has the lowest F1 score 0.1176. A complete table can be found in the Appendix.

We also analyzed the types of most common mistakes for every entity. There are several types of mistakes, which are the most frequent:

1. Nested entity recognition

One of the most common mistakes occurs when the model deals with nested entities, for example: “the Nobel Prize” is AWARD, but in some cases it can be presented as “Nobel” PERSON and “Prize” AWARD.

Sentence: “Нобелевская премия мира в 2017 году досталась организации « Международная кампания за запрещение ядерного оружия » ”.

True: []

Predict: [('нобелевская', 'PERSON')]

True: ['нобелевская премия', 'AWARD']

Predict: ['премия', 'AWARD']

However, in other sentences the span is extracted correctly:

Example: “Нобелевская премия мира 2018 года присуждена врачу Денису Муквеге из Конго и бывшей пленнице « Исламского государства » Езидке Наде Мурад из Ирака”.

True: [ (38, 41, 'нобелевская премия', 'AWARD'), (38, 42, 'нобелевская премия мира', 'AWARD')]

Predict: [(38, 41, 'нобелевская премия', 'AWARD'), (38, 42, 'нобелевская премия мира', 'AWARD')]

1. Not finding the correct start, end positions.

The entity type was predicted correctly, but the predicted value contains not only the entity itself, but also additional parts of the sentence.

For example: “5 августа российская постпанк группа « Буерак » из Новосибирска выпустила третии студийный альбом « Репост модерн ».”

True: [(38, 40, 'третии', 'ORDINAL')]

Predict: (38, 43, 'третии студийный', 'ORDINAL')]

1. Using the word's morphological paradigm.

For example, a word “треклист” (“list of songs”) is denoted as PROFESSION by the suffix “-ист”, which usually denotes professions (“пианист”, “альпинист”, “артист”).

1. Not taking into account a context.

For example: “отцом малыша является житель Симферополя 31 - летнии Олег, который заявил, что имя Биткоин выбрано не случайно.”

“True”: [36, 38, 'Биткоин', 'PERSON']

“Predict”: [(36, 38, 'Биткоин', 'MONEY')]

In this sentence “Биткоин” is a name of a child, but not the money.

And we have an opposite example, another sentence, where the same word was extracted correctly:

Sentence: “Биткоин Олегович родился весом в 3 кг и полностью здоров.”

Model true: [(15, 19, 'Биткоин Олегович', 'PERSON')]

Model predict: [(15, 19, 'Биткоин Олегович', 'PERSON')]

1. Classification errors between adjacent types.

For example: “за семь лет его фонд инвестировал около $ 200 млн в исследовательские проекты, разрабатывающие способы утилизации человеческих отходов”

Model true: [19, 22, 'за семь лет', 'TIME']

Model predict: [(19, 22, 'за семь лет', 'DATE')]

This type of errors can be explained by fuzzy separation of queries of adjacent types. And it means, that queries for adjacent entities should be described in more detail to avoid confusion, one way to achieve this can be to add new rules or to combine several methods of query generation (for example, annotation guideline notes + synonyms).

# Conclusion

In the current work, the possibility of using machine reading comprehension methods for solving problems of named entity recognition is investigated. Also, the MRC framework proposed by Lee et al. [ShannonAI., 2020], which uses BERT as a backbone, was trained on a modified Russian NEREL dataset.

The following can be noted as the results of this work:

1. We have demonstrated the efficiency of the method for the Russian language. To achieve this, the NEREL dataset was modified and trained. For evaluating the model’s efficiency there was chosen F1 score metric. F1 score for the NEREL dataset was 0.7721.

3. A table of F1 scores for all predicted named entity types, which are presented in the NEREL dataset, was implemented.

4. The main types of errors were analyzed.

At the next stage of the study, the following changes will need to be made:

1. Improve the description of requests for named entities, make a clearer distinction in the description of adjacent named entities.

2. Balance the number of named entity examples of all types to equally represent all entity types in the dataset.

One of the possible ways of further research is to conduct a more detailed study of nested entities in the Russian language.

# References

David Batista. 2018. Named-Entity evaluation metrics based on entity-level. https://www.davidsbatista.net/blog/2018/05/09/Named\_Entity\_Evaluation/

Denis Antukhov. 2020. Training a natural language model with BERT and Tensorflow. https://habr.com/ru/company/sberdevices/blog/527576/

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Ivan Smurov. 2019. NLP. Basics. Techniques. Self-development. Part 2: NER. https://habr.com/ru/company/abbyy/blog/449514/

Natalia Loukachevitch, Ekaterina Artemova, Tatiana Batura, Pavel Braslavski, Ilia Denisov, Vladimir Ivanov, Suresh Manandhar, Alexander Pugachev, and Elena Tutubalina. 2021. NEREL: A Russian Dataset with Nested Named Entities, Relations and Events. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), pages 876–885.

Nils Reimers, Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conferenceon Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, Pavel Kuksa. 2011. Natural language processing (almost) from scratch. The Journal of Machine Learning Research, 12, pages 2493–2537.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5849– 5859.

Zhu Chenguang. 2021. Machine Reading Comprehension. Algorithms and Practice. Elsevier.

# Appendix

The table for f1 score for predictions for every entity type

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Label** | **TP** | **possible** | **actual** | **precision** | **recall** | **f1** |
| FAMILY | 1 | 13 | 4 | 0.250000 | 0.076923 | 0.117647 |
| PENALTY | 7 | 18 | 14 | 0.500000 | 0.388889 | 0.437500 |
| FACILITY | 29 | 63 | 64 | 0.453125 | 0.460317 | 0.456693 |
| LOCATION | 24 | 61 | 41 | 0.585366 | 0.393443 | 0.470588 |
| DISTRICT | 11 | 25 | 17 | 0.647059 | 0.440000 | 0.523810 |
| WORK\_OF\_ART | 42 | 92 | 64 | 0.656250 | 0.456522 | 0.538462 |
| CRIME | 24 | 35 | 52 | 0.461538 | 0.685714 | 0.551724 |
| EVENT | 348 | 659 | 561 | 0.620321 | 0.528073 | 0.570492 |
| AWARD | 75 | 121 | 131 | 0.572519 | 0.619835 | 0.595238 |
| LAW | 35 | 62 | 55 | 0.636364 | 0.564516 | 0.598291 |
| DISEASE | 31 | 56 | 44 | 0.704545 | 0.553571 | 0.620000 |
| ORGANIZATION | 438 | 658 | 665 | 0.658647 | 0.665653 | 0.662132 |
| PRODUCT | 35 | 53 | 51 | 0.686275 | 0.660377 | 0.673077 |
| TIME | 31 | 47 | 43 | 0.720930 | 0.659574 | 0.688889 |
| NATIONALITY | 42 | 62 | 59 | 0.711864 | 0.677419 | 0.694215 |
| MONEY | 30 | 43 | 43 | 0.697674 | 0.697674 | 0.697674 |
| PROFESSION | 591 | 850 | 806 | 0.733251 | 0.695294 | 0.713768 |
| PERCENT | 5 | 7 | 7 | 0.714286 | 0.714286 | 0.714286 |
| STATE\_OR\_PROVINCE | 81 | 112 | 114 | 0.710526 | 0.723214 | 0.716814 |
| CITY | 195 | 238 | 298 | 0.654362 | 0.819328 | 0.727612 |
| IDEOLOGY | 32 | 43 | 43 | 0.744186 | 0.744186 | 0.744186 |
| ORDINAL | 76 | 106 | 91 | 0.835165 | 0.716981 | 0.771574 |
| LANGUAGE | 6 | 8 | 7 | 0.857143 | 0.750000 | 0.800000 |
| NUMBER | 176 | 224 | 196 | 0.897959 | 0.785714 | 0.838095 |
| PERSON | 825 | 955 | 993 | 0.830816 | 0.863874 | 0.847023 |
| DATE | 471 | 521 | 558 | 0.844086 | 0.904031 | 0.873031 |
| AGE | 120 | 146 | 128 | 0.937500 | 0.821918 | 0.875912 |
| COUNTRY | 408 | 456 | 449 | 0.908686 | 0.894737 | 0.901657 |
| RELIGION | 19 | 23 | 19 | 1.000000 | 0.826087 | 0.904762 |

The list of queries and corresponding labels

|  |  |
| --- | --- |
| "PERSON" | "Найди в тексте обозначение человека, человеческого существа, имени или фамилии", |
| "PROFESSION" | "Найти в тексте обозначение рода трудовой деятельности человека, обозначение труда или занятия, за которое человек получает оплату", |
| "ORGANIZATION" | "Найди в тексте обозначение производственного образования, сформированного из людей, деятельность которых координируется руководством для достижения общей цели", |
| "EVENT" | "Найди в тексте обозначение значимого явления, которое произошло в некоторый момент времени", |
| "DATE" | "Найди в тексте временную запись, включающую в себя число месяца, месяц или год", |
| "COUNTRY" | "Найди в тексте обозначение территории, имеющей политические, географические, культурные и исторические границы", |
| "CITY": | "Найди в тексте обозначение крупного населенного пункта, административного, торгового, промышленного и культурного центра", |
| "NUMBER" | "Найди в тексте обозначение количества или обозначение математической величины, при помощи которой производится счет", |
| "AGE" | "Найди в тексте обозначение возраста или обозначение количества лет", |
| "ORDINAL" | "Найди в тексте обозначение порядкового номера", |
| "AWARD" | "Найди в тексте обозначение приза, получаемого победителем соревнования, или обозначение награды за успехи", |
| "STATE\_OR\_PROVINCE" | "Найди в тексте обозначение территориальной административной единицы в пределах государства или страны", |
| "NATIONALITY" | "Найди в тексте обозначение принадлежности индивида к национальной или этнической группе, народности, нации, обозначение гражданства или юридической принадлежности к тому или иному государству", |
| "FACILITY" | "Найди в тексте обозначение коммерческого или административного здания", |
| "LAW" | "Найди в тексте обозначение правила или нормативно-правового акта, который принимается законодательным органом государственной власти и регулирует определённые общественные отношения", |
| "WORK\_OF\_ART" | "Найди в тексте обозначение произведения художественного творчества или иного материального продукта деятельности человека, имеющего определенную ценность", |
| "CRIME" | "Найди в тексте обозначение правонарушения или общественно опасного деяния, совершение которого влечёт применение мер уголовной ответственности", |
| "DISEASE" | "Найди в тексте обозначение расстройства здоровья, нарушающего деятельность организма или его отдельных органов", |
| "LOCATION" | "Найди в тексте обозначение места или местоположения какого-либо объекта или предмета", |
| "IDEOLOGY" | "Найди в тексте обозначение системы взглядов и идей, характеризующих какую-либо социальную группу, класс, политическую партию или сообщество", |
| "PRODUCT" | "Найди в тексте обозначение какого-либо предмета или результата человеческого труда, например, обработки, переработки, исследования", |
| "PENALTY" | "Найди в тексте обозначение меры воздействия на того, кто совершил правонарушение или преступление", |
| "TIME" | "Найди в тексте обозначение продолжительности, длительности чего-либо, измеряемого секундами, минутами, часами", |
| "MONEY" | "Найди в тексте обозначение меры стоимости товаров или услуг, используемой для обмена", |
| "DISTRICT" | "Найди в тексте обозначение административно-территориальной единицы или обозначение части города и страны", |
| "RELIGION" | "Найди в тексте обозначение системы взглядов, обусловленной верой в сверхъестественное, включающей в себя свод моральных норм и обрядов, и объединяющей людей в институты", |
| "PERCENT" | "Найди в тексте обозначение одной сотой части чего-либо, используемой для обозначения доли чего-либо по отношению к целому", |
| "LANGUAGE" | "Найди в тексте обозначение сложной знаковой системы, естественно или искусственно созданной для коммуникации между людьми", |
| "FAMILY" | "Найди в тексте обозначение основанной на браке или кровном родстве малой группы людей, члены которой связаны общностью быта, взаимной ответственностью и взаимопомощью" |