Named Entity Recognition for Russian Judicial Rulings Text

Maria Averina, Olga Levanova, Natalia Kasatkina
P.G. Demidov Yaroslavl State University,
Yaroslavl, Russia
maverina518@gmail.com, olaydy@gmail.com, ninet75@mail.ru

***Abstract*—The article presents the solution of named entity recognition problem for legal Russian-language texts. We studied CRF, LSTM, BERT and BiLSTM and their combinations. The models were tested with various parameters of text preprocessing and words vector representations. The best result was shown by fastext vectorization with BiLSTM and CRF model, the value** $F-score$ **is** 0.86**.**

I. Introduction

The extracting information problem often arises while processing texts. For example, in informative search it is necessary to find a r esource t hat m atches t he q uery criteria. Also, there is a task of automatic selection of news related to certain events by place, time of action, event participant, etc. The task of named entity recognition (NER) is the automatic identification o f t ext f ragments w ith a ppropriate meaning"— named entities (NEs). Typically, named entities are chosen from a set of task-specific s emantic c ategories. F or example, for news articles the established classes are person (PER), location (LOC), organization (ORG) and others (MISC) [1]. It is essential to highlight information about persons, places and organizations in a text.

For other subject areas the entities may be less generalized, and more complex. In business, the task of extracting certain entities from official d ocuments o ften o ccurs. F or example, after scanning a document in the bank, it is useful to automatically determine not only the full name of a client, but also the employee, date, passport details, credit amount etc. Based on this information, business processes, such as automatic data validation, adding to the database, and other tasks can be carried out. Therefore, selection of semantic entities from a text is a very important business task.

This article aims at solving the problem of named entities recognition in court documents. Court rulings in Russian language were chosen as the subject of the study. Court documents are not similar to each other, and differ from texts in other domains. The bank documents typically are well structured, and we are interested in a more general case for a universal solution. This work studies the NER task in weak-structured Russian legal texts.

Recently, approaches based on machine learning have been actively developed for the NER task. Pre-trained transformer architecture BERT, recurrent neural networks are frequently used nowadays. We use LSTM, BiLSTM architecture and their combinations with CRF, as well as the CRF, BERT and test them.

The quality of the trained model depends on the text preprocessing, model architecture and its internal parameters. Each architecture was tested with different sets of internal parameters (depending on the method) and various text preprocessing options. We investigated state-of-the-art methods for NER tasks, chose the best models for different entities and combined into one solution.

This article deals with the NER problem for the Russian texts, and discusses approaches to solving this problem. The task of extracting named entities is especially relevant for the Russian language, as almost all existing systems and libraries work successfully with the English texts as for languages other than English, results are significantly worse.

II. Related works

The most research for the NER task is devoted to the processing of English texts. Text corpus based on news articles with classical entities generally appears in articles. There are just a few studies for other entities and text types. To solve our problem, a regular degeneracy approach is frequently used. This approach is not flexible, therefore, i n t he m odern world, machine learning is increasingly used instead of it.

here are state-of-the-art methods such as CRF, LSTM, BiLSTM and BERT [2] for NER tasks. For news articles in English, the model achieved a quality of about 0.90 F1. For example, Arda Akdemir and others [3] provided a decision with quality F1 0.93 ± 0.04. The multilingual solutions show inferior quality. Jana Strakov, Milan Straka and Jan Hajic [4] showed diverse quality iin English (0.93), German (0.88), Dutch (0.92) and Spanish (0.88).

Deeppavlov [5] solved the problem of classical entities (PER, LOC, ORG, MISC) in the Russian language. The authors used deep neural network models BiLSTM and their combinations with CRF. All models were evaluated on three datasets: Gareev dataset, Person-1000 and FactRuEval-2016. The best scores across all data for Person and Organization are 0.95 and 0.84. In 2019, Deeppavlov [6] conducted a study on a multilingual model in Russian, Vietnamese, English and Chinese. The authors obtained F1-scores 0.91, 0.94, 0.91, 0.92

respectively. Testing was performed on the following datasets: Gareev, VLSP-2016, CoNLL-2003, and MSRA.

The quality of recognition non-classical entities is significantly less than for classical entities. Similar task for Dutch court rulings was solved by Simon Brugman [7]. In this article, the NER results were used to anonymize documents (removing personal information). The best result was obtained by applying the LM- BiLSTM-CRF model (0.87). In a similar study [1] Elena and Rehm Leitner, Georg and Moreno-Schneider Julian are engaged in anonymization of German courts text, and the best quality was achieved by the BiLSTM-CRF+ model (0.95).

There are just a few solutions for non-classical entities recognition in the Russian language and they show poor results. For example, there is a solution by Alexander Sboev, Sanna Sboeva and others [8]. In research they achieved a 0.61 by F1 score.

III. Problem statement

In this paper, we consider the solution of NER problem for court records. Specificity o f t he t ask r equires t hat categories reflect t ypical e ntities f or t hese d ocuments (court decisions). For example, we need to pick from a text some information going beyond “persons” (e.g. defendant, plaintiff or judge). For our research we used an open judicial statistics database [9] containing court rulings as it contains a large number of available and relevant documents. Namely, these texts contain many different names, dates, denominations, amounts, etc.

№ <*doc num, 2-1606/2018*>

РЕШЕНИЕ
Именем Российской федерации
<*date court, 02 ноября 2018 года*> <*court, Кировский районный суд г.Томска*> в составе:
председательствующего судьи <*judge, Алиткиной Т.А.*>,
при секретаре Бондаревой Е.Е.,
с участием представителя процессуального истца старшего прокурора Кировского района г.Томска Морарь И.В., материального истца <*plaintiff, Полищука Э.Г.*>, представителя ответчика <*defendant, Семенова С.М*>., действующего на основании доверенности от 17.05.2017 (срок действия доверенности три года),

…

собственному желанию, взыскании задолженности по заработной плате, компенсации, предусмотренной ст.236 ТК РФ, компенсации за задержку выплаты заработной платы, взыскании денежной компенсации морального вреда <*court decision, удовлетворить частично*>.

Fig. 1. Example from the dataset. Beginning of text

Experts marked up 344 text files using the BRAT [10] tool. Fig. 1 and Fig. 2 provide examples of a document markup. We played the entity names and their values up in bold type. All entities can be conditionally divided into groups by complexity in recognition. The group of simple entities includes document number (”doc num”), court decision, judge (Fig. 1).

Also, there are complex entities, containing more than two words, such as ”appeal time”, ”plaintiff”, ”payment fine”, ”payment amount”, ”defendant” and ”court”. Average number of the words for each entity is represented in the first column Table I. Last column represents the percentage of documents containing relevant entities. In the last column, we can see that the number of entities significantly differs in the dataset. So, the solution can be sensitive to objects amount and have different qualities for entities.

Moreover, some entities have different meanings depending on a document. For example, the defendant can be a person,

<*payment fine, задолженность по заработной плате*> за период с 26.12.2017 по 22.01.2018 в размере <*payment amount, 20 520,00 руб.*>; <*payment fine,денежную компенсацию, предусмотренную ст.236 ТК РФ*>, в размере <*payment amount,796,20 руб.*>; <*payment fine,утраченный заработок за время вынужденного прогула*> в размере <*payment amount, 91 198,80 руб.*>, денежную <*payment fine,компенсацию морального вреда* >в размере <*payment amount, 3 000 руб*>.

…

Решение может быть обжаловано в Томский областной суд путем подачи апелляционной жалобы через Кировский районный суд г.Томска <*appeal time, в течение 1 месяца со дня изготовления решения в мотивированном виде*>.

Судья: (подпись) <*judge, Т.А.Алиткина*>

Fig. 2. Example from the dataset. End of text

TABLE I. Entity statistics in the dataset

entity	size entity	percentage of documents with entity	number of entities in data
plantiff	2,4	98%	338
payment fine	2,4	77%	265
payment amount	8,2	70%	241
judge	1,9	98%	337
doc num	1	78%	268
defendant	6,0	98%	339
date court	2,6	93%	320
court	4,2	98%	238
court decision	1,9	94%	323
appeal time	9,6	95%	327

an organization, or a defendant’s representative. The entity ”payment fine” i n d ifferent d ocuments m ay h ave a variety of meanings such as law number, payment or compensation. Complex entities can be discontinuous, for example, in Fig. 2, the entities ”payment fine” and ”payment amount” alternate. Therefore, these entities are more difficult to find.

Two approaches to quality calculation for NER: *F1-measure* and standard *F1-score* were proposed at the CoNLL [2] conference. Since we have discontinuous entities in our data, we decided to use F1-score. This approach gives a little bit more optimistic results, but testing has shown that the metrics do not differ much (0.04 ± 0.02).

So, to solve our task we divided it into steps. The first step in this research is to select various features of words and their context. We investigated features based on regular expression and various algorithms of words vector representation. For NER task we trained different models: CRF, RNN, combinations BiLSTM and BERT with CRF. Finally, we selected the best models for each entities by F1-score. Also the training and prediction times were evaluated for different models and feature vectors.

IV. Feature extraction

For text processing tasks the problem of the features’ choice remains an open question. There are some approaches based on regular expressions, morphological features, syntactic and

semantic analysis. The most obvious solution is the extraction of information by using regular expressions. Thus, it is possible to extract information using nearest punctuation marks or letter cases. For example, the document number "№ 11255588," is divided into "№" and "11255588,", where the № is identified as a special character word. There are special characters used in this work: $@, \#,$ №, $\backslash, \%, \$, |$.

Here is the list of features based on regular expressions:

- first letter is uppercase;
- first letter is small;
- letters are small;
- letters are capitalized;
- presence of @ inside a word;
- the presence of a comma and (or) dot at the end (beginning) of a word;
- presence of digits in the word.

A "word" can be used as a feature, but it has no special information for the model. Thus, if the same organization name or a surname is often mentioned in the training set (e.g. judge name in court documents), then the algorithm "hooks" to that concrete word.

Words differ in grammatical case, gender, part of speech and other grammatical categories. It complicates the work with algorithms. One way to solve this problem is the word lemmatization or stemming meaning the words reduction to the initial form. In this case, part of speech, number and gender can be used as features. Note that a unique value is given to every character. For example, for the % character the morphology feature value would be PERCENT SYMBOL. We can also remove stop words from the text, because they make the text noisy [11].

Another common approach to feature computing is a word vectorization meaning the representation of words as a vector of numbers (word embedding). For the vector words representation the Word2Vec [12] and FastText [13] algorithms were taken. To improve the quality of entity recognition, it is good to take into account the word context. To follow this principle we used features of words neighbors, as words' characteristics.

V. Methods

CRF. The most popular algorithm for solving the named entity recognition problem is conditional random fields (CRF). This method optimizes the entire token chain and takes into account any dependencies in data. It also works well with recurrent neural networks. CRF is good for solving segmentation and sequence marking problems, for example, automatic extraction of keywords from texts, extraction of named entities (entities classification), s entiment a nalysis or automatic speech recognition.

The CRF learning process has a large computational complexity equal to $O(mNTQ2nS)$:

- *m*"— number of training iterations;
- *N*"— number of training sequences;
- *T*"— average training sequence length;
- *Q*"— number of output classes;
- *n* "— number of features in the training matrix;
- *S* "— optimization algorithm running time at each step.

The complexity of model prediction is slightly less than model training. It is equal to $0(K|C|^3)$, where K is the number of input data sequences, C is the number of possible output classes. In practice, the time complexity of CRF training is higher due to overhead computation.

LSTM and BiLSTM. The second most popular approach is recurrent neural networks (RNN). A recurrent neural network encodes an input sequence of words into a context-sensitive representation. LSTM and BiLSTM architectures are usually used for NER problem.

The input of recurrent neural networks is a sequence. Each element in a sequence is characterized by a vector of numbers. We can process documents in different ways: full document, page by page, in separate paragraphs or sentences. By separately processing each paragraph the information about relationship to the nearest paragraphs is lost. The processing can occur while maintaining a structure of documents, pages, paragraphs or sentences. So, we can compose sequences according to document structures.

For word vector representation we used the following algorithms: fasttext (f), word2vec (w), bert embedding (bert) models [14] and "bag of words" [15] with an Embedding layer. Fasttext and word2vec models are based on the contextual similarity of words. To reduce vector space it is recommended to use word lemmatization or stemming. Moreover, the authors implemented the possibility of adding information about morphological and regular features in vector representation.

Unlike LSTM, the Bidirectional LSTM model propagates signals both backwards and forwards ("two-way attention"). Thus, the model takes into account previous and subsequent words.

BiLSTM + CRF. One problem of CRF is capable of capturing the dependencies between labels in the forward direction only. It can be resolved by introducing a BiLSTM between inputs and CRF (fig. 3) [16]. Combination of CRF model with LSTM or BiLSTM neural network output weights should improve the accuracy of the extracted named entities.

BERT. Today, BERT [17] is often used for natural language processing. It shows good results for various tasks. BERT is based on the encoder-transformer architecture. This architecture was invented as an alternative to computationally expensive recurrent architectures. In each of its encoder layers, "two- way attention" is applied. This allows context on both sides of the token. Therefore, model determines the tags for tokens more accurately. BERT architecture implies the ability to train from scratch or fine-tune a pre-trained model.

BERT pre-training requires a large text corpus and large computing power. The developer community shared a pre-trained model for different languages and different NLP tasks. Fig. 4 shows the BERT architecture for the NER problem. In text processing, we can take into account the case of letters, so we implemented two modifications of BERT: case sensitive and case insensitive. Moreover, BERT produces two models

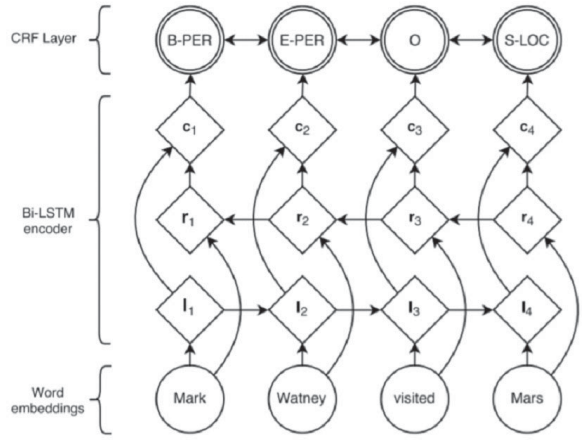

Fig. 3. BiLSTM architecture with CRF

different in size: BERT BASE (basic), BERT LARGE (advanced).

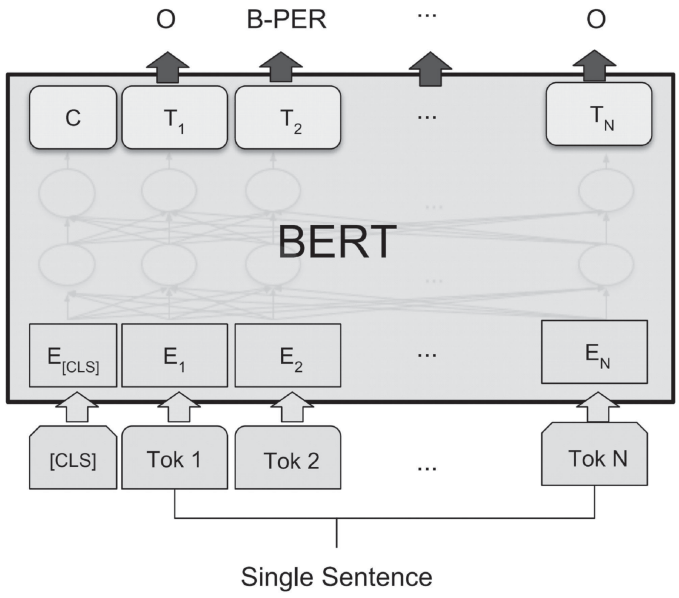

Fig. 4. BERT NER architecture

BERT Embedding. BERT embedding is an approach to word representation. It is based on BERT model and can be used with recurrent neural networks. The word sequences are passed on to Bert Tokenizer, where each token is replaced with an ID from a lookup table (supplied with a pre-trained model). Then this vector representation is fed to the input of recurrent neural network.

VI. Experiments

Presented methods for solving NER problem were tested on the dataset of 344 court records. The data was split to train and test a set by ratio of 80 and 20 percent respectively. All results were evaluated on a test set. Each model was tested with different parameters values and text preprocessing options. The tables below demonstrate the best results.

Models were implemented on Python 3.8. We used popular libraries like Tensorflow 2 , K eras f or L STM a nd biLSTM, Pytorch-transforms for BERT, CRF from sklearn library, crfsuite and keras-contrib. For feature extraction we used NLTK, pymorphy2, gensim.

CRF. For the CRF model, we need to choose internal parameters. The basic parameter is the optimization algorithm and we can choose it as: L-BFGS Gradient Descent (lbfgs), Stochastic Gradient Descent (l2sgd), Averaged Perceptron (ap), Passive-Aggressive (PA, pa), adaptive weight vector regularization (AROW, arrow). Algorithms *lbfgs* and *l2sgd* have regularization parameters *L1* and *L2*. Both regularizations can be used at the same time just for *lbfgs*.

Different features can be used for CRF model, such as regular expression (r), word value (v), part of speech (m), lemmatization (l), Word2Vec (w), FastText (f). In addition, the features of neighboring words also characterize the token well. The number after the feature name means the number of neighbors on the left and right from the current word. For example, $f2$ means that five FastText features were used: for the current word and by two adjacent words on each side.

TABLE II. Comparative analysis CRF on different feature sets

entity	r1, v1 pa	r3, v3 pa	r3, v3, m3 pa	f3(10), r3, m3 lbfgs
appeal time	0.92	**0.93**	**0.93**	0.92
court	0.94	**0.97**	**0.97**	0.91
court decision	**0.71**	**0.71**	**0.71**	0.60
court date	0.89	**0.94**	0.92	0.82
defendant	0.52	0.61	**0.67**	0.57
doc num	0.95	**0.98**	0.97	0.95
judge	**0.97**	**0.97**	**0.97**	0.95
payment amount	0.64	0.73	**0.77**	0.72
payment fine	0.64	**0.69**	0.68	0.55
plaintiff	0.85	**0.86**	**0.86**	0.82
F-macro	0.82	0.85	**0.86**	0.80

The results of CRF model testing on various feature sets are shown in Table II. Note that changing the number of neighbors from 1 to 3 increases quality (columns 1 and 2). By adding feature $m3$ we improve the quality. It is evident from "payment amount" (discontinuous) and the "defendant" entities. Word value is not a good feature due to the reasons discussed above. However, the quality decreased due to the replacement of word value $v3$ to FastText $f3$.

Thus, the smallest $F1$ spread is observed on $(r3, v3, m3)$ with the optimizer *pa*. Also, the maximum average result is achieved by application of this feature set. The entity "defendant" had shown the worst recognition quality. Apparently, this is due to the variety of entity meanings in different documents (person or representative). Moreover, the

quality for simple and complex is better than for discontinuous ones (payment fine/amount).

LSTM, BiLSTM, their Combinations with CRF and BERT. We need to choose the method of words representation. The embedding layer can be selected from fasttext, w2v, bert embedding and "bag of words" (BOW). Note that the fasttext and w2v were trained on our corpus, and for bert embedding we did not use fine-tune. The testing showed that LSTM architecture works worse than BiLSTM.

The results of the BiLSTM model with different embeddings are shown in table III. The worst quality was obtained by using a bag of words (the simple method). Also, BERT showed bad results. It is especially visible on the court's decision entity. Probably, it is due to the insufficient size of the training set.

Fasttext and Word2vec gave approximately the same results, but Word2Vec is inferior to Fasttext by the majority of entities. Thus, the best quality of BiLSTM has been achieved by using the fasttext method.

TABLE III. COMPARATIVE ANALYSIS OF VECTOR REPRESENTATIONS FOR BILSTM

entity	Fasttext	Word2Vec	Bert embedding	BOW
appeal time	0.78	**0.79**	0.74	0.78
court	0.79	0.74	**0.80**	0.69
court decision	0.49	**0.53**	0.13	0.41
court date	0.75	0.71	**0.78**	0.65
defendant	0.45	**0.52**	0.50	0.43
doc num	**0.75**	0.73	0.74	0.55
judge	**0.82**	0.77	0.65	0.56
payment amount	**0.61**	0.59	0.48	0.45
payment fine	**0.55**	0.52	0.34	0.37
plaintiff	**0.68**	0.67	0.47	0.50
F-macro	**0.66**	0.65	0.60	0.53

Generally, the LSTM and BiLSTM architectures performed poorly compared to CRF. The average F-measure does not exceed 0.7. Through numerous experimental tests, we found out that to improve the quality of BiLSTM + CRF model, we should:

- save document structure;
- remove stop words;
- use stemming and lemmatization.

The results obtained at different models trained with best parameters are presented in table IV. By comparing the second and the fourth columns, we can see that the CRF model is inferior to BiLSTM with CRF (trained on the similar feature set). The BERT model is not efficient especially for complex entities; see the third column. Probably, it can be explained by an insufficient training set. Simple entities are recognized better than complex ones. The value of F1-score for discontinuous entities is much worse. The best results demonstrated by the CRF model and BiLSTM + CRF, the F-macro are 0.86 and 0.85 respectively.

For each entity we can select the best model. In the table values are highlighted in gray. We choose the best results for each entity, and combine corresponding models in a final

TABLE IV. COMPARATIVE ANALYSIS OF THE BEST MODELS

entity	CRF r3,v3, m3	BiLSTM and CRF f, r, m, l page	BERT page stop words	CRF f3, r3, m3
plaintiff	0.82	**0.86**	0.69	0.82
payment fine	0.68	0.60	0.53	0.55
payment amount	0.77	0.65	0.55	0.72
judge	0.97	0.91	0.82	0.95
doc num	0.98	0.91	0.69	0.95
defendant	0.67	**0.71**	0.57	0.57
date court	0.92	**0.96**	0.79	0.82
court	0.97	0.96	0.85	0.91
court decision	0.71	**0.79**	0.48	0.60
appeal time	0.93	**0.95**	0.82	0.92
F1-macro	**0.86**	**0.85**	0.74	0.80

solution. Thus, the quality of the final m odel w ill b e better than any model we have considered.

Model Prediction Error Analysis. Below we will be discussing the errors of top CRF and BiLSTM + CRF models prediction in detail. We chose the documents with wrong predictions and analyzed errors visually. In Fig. 5 - 8, the marked up entity is framed by a tag (the entity name) and words predicted by the model are highlighted in gray.

Дело №***<doc num>***2-1956/2018***<doc num>*** г. ...
РЕШЕНИЕ
Именем Российской Федерации
<date court>30 октября 2018***<date court>*** года г. Пенза
<court>Первомайский районный суд г. Пен-зы***<court>*** в составе:
председательствующего судьи***<judge>*** Гошуляк Т.В***<judge>***,
при секретаре Беспаловой К.И.,
с участием прокурора Ермаковой И.В.,

Fig. 5. Example of document with omitted entities parts

Analysis has shown that the most typical mistake is incorrect detection of beginning or ending of an entity. Fig. 5 shows that in the case of entities with a person's name, the initials are often omitted. When a model recognizes a date, it mistakenly loses the number although the month and the year are found correctly. Sometimes the model skips the word in the middle of the entity (like for court in Fig. 5).

рассмотрев в открытом судебном заседании в здании суда гражданское дело по исковому заявлению ***<plaintiff>***Плотниковой И.Н***<plaintiff>***. к ***<defendant>*** ООО «Теплоцентраль»***<defendant>*** о восстановлении на работе, взыскании заработной платы за время вынужденного прогула, компенсации морального вреда,
...

Fig. 6. Typical error for "defendant" entity

We received the worst recognition quality for entities with different meanings. The error analysis showed that the entity "defendant" in the meaning of a "person name" is recognized

well. In the ”organization” sense the model sometimes does not predict all the words. For example (Fig. 6), it almost always detects the abbreviation "ООО", but often omits the name of the organization.

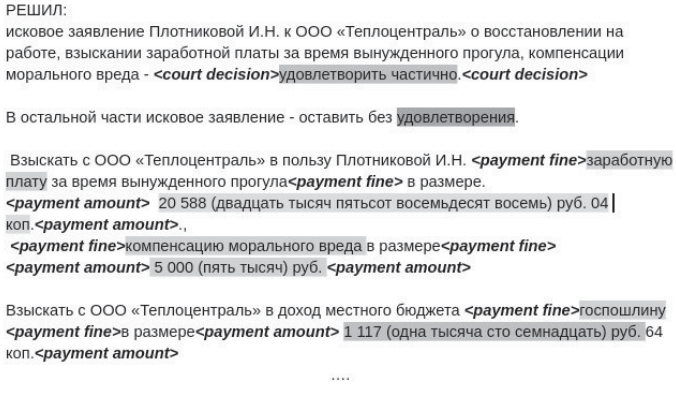

Fig. 7. Error for payment amount and payment fine and court decision entities

Court documents are poorly structured. The text contains the beginning part (Fig. 5) then an arbitrary part with a description of the lawsuit meaning, case details and justification of the court’s decision. In the end, each document contains the conclusion. There is a text fragment with the final formulated court’s decision, starting with the word "РЕШИЛ:" (Fig. 7). Some entities, for example plaintiff’s name, occur many times in different places in document (Fig. 6 and 7). To avoid repetition of information, these entities were marked up only in the final part.

In the Fig. 7 we can see that the entities ”payment amount” and ”payment fine” are discontinuous (occur more than once in the markup). These entities are more difficult to detect and model sometimes recognizes not a full entity. It deciphers some parts. The most difficult task for the model is to detect the ”payment fine” because it has a big value variety. The ”payment amount” always contains numbers and it makes it easy to find it. Note that prediction results of BiLSTM + CRF model are represented in Fig. 7, but CRF showed better quality for these entities.

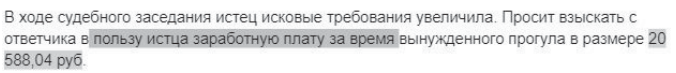

Fig. 8. Error for payment amount and payment fine entities

Another common error recognition is when entities are detected in wrong places. For example, the “court decision” entity often takes a value ”satisfy partially” (Fig. 8). Apparently during training, the model memorizes the word *satisfy* and being oriented on it detects this word in other places. We provide an example of the text from an unstructured part of the document (see Fig. 8). We can see that amounts are predicted as ”payment amount” in other parts of the text. Also the ”payment fine” has been detected in the wrong place of a document, but the meaning turned out to be correct.

During the error analysis we found that CRF model skips whole entities more often than BiLSTM + CRF (recognizing at least one part of an entity). So, we decided for each entity choose the best model and show it prediction results.

VII. Discussion

The final goal of the study is to develop an application for automation of business processes related to processing of text documents. For modeling business processes, the text file was converted into a pdf-scan, after that we used text recognition using OCR API Tesseract (Optical Character Recognition). Inevitably OCR algorithm makes some errors leading to text distortion. This can affect the quality of an entity recognition. Our testing showed that the quality of a model on source txt-files was better by 0.03 ± 0.02 (F1-score) than pdf-scan.

Our research is aimed at creating a universal model which suits different documents and entities. That is why, the NER tasks were tested on weak-structured Russian legal texts. For each entity, the model and its parameters were selected to show the best value for F1-score, and these models were integrated in one solution.

Note that this solution was tested on a credit contract documents dataset of 200 structured texts. The testing on this corpus gave very good results, most entities had no errors during recognition, the worst quality of 0.92 was shown by the ”credit contract date” entity. We got better results on smaller text corpus because credit contract documents are better structured and comprise simple entities.

TABLE V. Training time of models

model	train time
CRF (r3,v3) pa	0:21:48
Fasttext, BiLSTM + CRF	2:35:15
BERT	5:15:10
Bert embedding, BiLSTM + CRF	3:47:40
CRF, (r3,v3), lbfgs	6:09:32
Word2Vec	0:00:31
Fasttext 10	0:00:41

We also discussed the time of training models and predicting. The CRF models were trained on a CPU (Intel(R) Xeon(R) CPU E5-2698 v4 2.20GHz), other models — on a GPU (Tesla V100-SXM2 16GB). The first rows of Table V show the training time of the best models from Table IV. As we can see in line 4, the BERT has the longest learning time and it takes all computing power. The BERT Embedding with BiLSTM + CRF needs much less time than the BERT model. The training time for the CRF strongly depends on the optimizer and dimension of feature vector. Under the same conditions the training time with optimizer *pa* less by 10 times than *lbfgs*. Note Word2Vec and Fasttext have a very small learning time.

In practice, the more important thing is to estimate the time of model prediction, the best models being shown in

table VI. The complication of the model architecture leads to an increase in prediction time significantly. The CRF model prediction takes less time — 2 seconds. The model BiLSTM + CRF works 3 times slower. The BERT Embedding with BiLSTM + CRF needs much more time than the BERT model.

TABLE VI. TIME OF MODELS PREDICTION

model	predict time (sec)
CRF (r3,v3) *pa*	**2**
Fasttext, BiLSTM + CRF	6
BERT	3.7
Bert embedding, BiLSTM + CRF	18.2
CRF, (r3,v3), lbfgs	**2.2**

There are many directions for further research: development of new informative features; testing of Glove model; usage of Fasttext and Word2vec pre-trained on other big corpora, training all models (especially BERT) on extended data set. Also it is interesting to apply RuBERT instead of multilingual pre-trained BERT. The perspective research direction is the use of alternative models, for example, BERT+CRF.

VIII. CONCLUSION

In this article we present a solution to the NER problem for Russian court rulings. We researched solutions based on the CRF, LSTM, BiLSTM, LSTM+CRF, BiLSTM + CRF and BERT architectures.

All models were tested with various parameters of text preprocessing and vector representation of words. For RNN the Fasttext was selected as the best vector representation algorithm. The quality of LSTM and BiLSTM architectures turned out unsatisfactory (F-measure less than 0.6). The BERT model has not given good results probably due to the small training dataset.

Through numerous tests we have concluded to use the combination of BiLSTM and CRF, which showed the best results, F-score equals to 0.86. It is worth noting that BiLSTM and CRF more often define at least part of a named entity, unlike CRF model. Comparing the training and prediction times of the two best models showed that CRF is several times faster.

ACKNOWLEDGMENT

The reported study was funded by YSU Programme according to the research project № P2-K-1-G-5/2021.

REFERENCES

[1] E. Leitner, G. Rehm, and J. Moreno-Schneider, *Fine-grained named entity recognition in legal documents*. Springer, 2019.

[2] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 50–70, 2020.

[3] A. Akdemir, A. Hürriyetoğlu, E. Yörük, B. Gürel, Ç. Yoltar, and D. Yüret, *Towards generalizable place name recognition systems: anal-ysis and enhancement of NER systems on English News from India*, 2018.

[4] J. Straková, M. Straka, and J. Hajic, "Neural architectures for nested NER through linearization," *CoRR*, 2019.

[5] A. L. The, M. Y. Arkhipov, and M. S. Burtsev, "Application of a hybrid bi-lstm-crf model to the task of russian named entity recognition," *CoRR*, vol. abs/1709.09686, 2017.

[6] M. Y. Arkhipov, M. S. Burtsev *et al.*, *Application of a Hybrid Bi-LSTM-CRF model to the task of Russian Named Entity Recognition*, 2017.

[7] S. Brugman, *Deep Learning for Legal Tech: exploring NER on Dutch court rulings*. Data Science Group Faculty of Science, 2018.

[8] A. Sboev, S. Sboeva, I. Moloshnikov, A. Gryaznov, R. Rybka, A. Naumov, A. Selivanov, G. Rylkov, and V. Ilyin, "Analysis of the full-size russian corpus of internet drug reviews with complex ner labeling using deep learning neural networks and language models," *Applied Sciences*, vol. 12, no. 1, p. 491, 2022.

[9] *IEEE official website, Manuscript Templates for Conference Proceedings*. [Online]. Available: http://www.cdep.ru/index.php?id=79

[10] *Brat rapid annotation tool*. [Online]. Available: http://brat.nlplab.org

[11] V. A. Kozhevnikov and E. S. Pankratova, "Research of text preprocessing methods for preparing data in russian for machine learning." *Theoretical & Applied Science*, no. 4, pp. 313–320, 2020.

[12] K. W. Church, "Word2vec," *Natural Language Engineering*, vol. 23, no. 1, pp. 155–162, 2017.

[13] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the association for computational linguistics*, vol. 5, pp. 135–146, 2017.

[14] H. Wu, J. Ji, H. Tian, Y. Chen, W. Ge, H. Zhang, F. Yu, J. Zou, M. Nakamura, J. Liao *et al.*, "Chinese-named entity recognition from adverse drug event records: Radical embedding-combined dynamic embedding–based bert in a bidirectional long short-term conditional random field (bi-lstm-crf) model," *JMIR medical informatics*, vol. 9, no. 12, p. e26407, 2021.

[15] K. Irie, R. Schlüter, and H. Ney, *Bag-of-words input for long history representation in neural network-based language models for speech recognition*, 2015.

[16] R. Panchendrarajan and A. Amaresan, *Bidirectional LSTM-CRF for named entity recognition*, 2018.

[17] K. Labusch, P. Kulturbesitz, C. Neudecker, and D. Zellhöfer, *BERT for Named Entity Recognition in Contemporary and Historical German*, 2019.