Feature Extraction Method from Electronic Health Records in Russia

Denis Gavrilov, Alexander Gusev, Igor Korsakov
K-SkAI
Petrozavodsk, Russia
dgavrilov@ webiomed.ai, agusev@webiomed.ai,
ikorsakov@webiomed.ai

Roman Novitsky, Larisa Serova
K-SkAI
Petrozavodsk, Russia
roman@webiomed.ai, lserova@webiomed.ai

Abstract. The medical language is the basis of the electronic medical records (EHR), and up to 70 percent of the information in these records were writing in natural language, in the free text part [1]. The last few years have seen a surge in the number of accurate, fast, publicly available name entity recognition (NER) parsers [2,3]. At the same time, the use of NER parsing in natural language processing (NLP) applications has increased. It can be difficult for a non-expert to select a good "off-the-shelf" parser. We present a method of using statistical NER parsers on a medical corpus of Russian. We developed a new tool that gives a convenient way to extract NER from unstructured medical documents.

I. Introduction

Medicine benefits greatly from deep learning because of the huge amount of data generated (150 exabytes in the United States alone, which is 48% increase per year), as well as the growing proliferation of medical devices and digital recording systems.

Deep learning models scale to large data sets, partly because of their ability to run on specialized computing hardware, and partly because they continue to improve with large amounts of data, allowing them to surpass many classic machine leaning (ML) approaches. Deep learning systems can accept several types of data as input, due to the special relationship to heterogeneous health data. The most common models are trained using controlled learning, in which data sets consist of anthropometric data (for example, height and weight of patients) and corresponding output data tags.

Natural language processing (NLP) focuses on analyzing text and speech to derive meaning from words. Recurrent neural networks (RNNs)-deep learning algorithms, which are effective in processing sequential input data such as language, speech, and time series data - play an important role in this area. Notable successes of NLP include machine translation, text generation, and image caption [1].

In healthcare, consistent in-depth learning and language technologies enable their application in areas such as electronic health records (EHR). EHR of a large medical organization can record medical operations of more than 10 million patients over a decade. Hospitalization alone usually generates 120,000 units of data. The potential benefits derived from these data are significant. Even though machine learning technologies have significantly improved the accuracy and quality of medical staff work, there are still many problems in ML processing in medicine, including insufficient structured data available for training. However, unstructured data does not allow its use in training — it's needed to create datasets of structured data. There are two ways to create a dataset: to conduct a study for 12-22 years to collect data from patients, and then-to make an analysis and make a limited set of structured data, including laboratory results, vital signs, diagnostic tests, and demographic data. Example of the first approach is Framingham Heart Study, which has been held for more than 70 years [4]. And the second one is using NLP to extract features from medical documents, such as an EHR [5].

One of the most interesting applications of artificial intelligence (AI) is predictive medicine. Data science predictive analytics methods learn from historical data and make accurate predictions about future results. They process patient data, make sense of clinical observations, find correlations, symptom associations, familiar backstories, habits, and illnesses, and then make predictions. The influence of certain biomedical factors, such as genome structure or clinical variables, is considered when predicting the development of certain diseases. Common cases include prognosis of disease progression or prevention to reduce risk and negative outcomes. The main advantage is to improve the quality of life of patients and improve the working conditions of doctors. Modern deep learning methods model the temporal sequence of structured events, which occurred in a patient's medical history using convolutional and recurrent neural networks to predict future health events.

II. Main part

A. Medical data

Despite the efforts made to structure the clinical narrative, the fact that structured representations are not able to provide the level of description and convenience needed by the Clinician means that unstructured natural language still prevails in the medical record. Indeed, many important observations remain unregistered in a structured record, appearing only in free text stored next to empty fields and forms. Free text is convenient for expressing clinical concepts and events, such as diagnosis, symptoms, and interventions [5]. The structured data is freezing clinical language and has limitations [6]. Most medical language is nuanced and uses negation, time expressions, and hedging phrases extensively. One way to resolve this contradiction is through linguistic analysis of free text, a field of computer science known as

NLP. The NLP of health records is almost as old as the computerization of these records.

NLP does not provide a detailed answer to the problem of extracting medical information from natural language texts, although the very reason why clinicians value language is an accurate description of the disease, which of course makes it difficult to extract medical information.

B. Dataset

We retrieved anonymized data from three hospitals in Russia. These data as a text file was downloaded from clinical decision support system (CDSS) “Webiomed” that was more than a million characters long. This set is including data of non-overlapping sentences: training set 2017, 1123 development and 868 testing.

We specifically selected patients with cardiovascular diseases and extracted medical records of patients aged 40 to 80 years, which contain unstructured text with information related to the patients' condition [7].

To create a dataset for training mathematical models, the information stored in an electronic health record and/or electronic medical history is used. We plan to extract the following information about patient: height, weight, blood pressure: systolic and diastolic, breathe rate & heart rate. Some fragments of blood pressure usage in electronic health records in Russian show in Fig1. To understand how to extract these features we collect sentences with them.

болен около года, появились боли в шейном отделе, лечился с помощью ипликатора, , ухудшение с февраля 2014г, гипертонический криз, повышение **ад до 180\\110мм рт ст**, проводилось амбулаторное лечение у терапевта, ад снизилось, но головные боли остались, продолжил лечение у невролога, к труду приступил с 13.03.14 эффект от лечения

Fig. 1. Some fragments of blood pressure usage in electronic health records in Russian

One of the most well-known methods of extracting information until recently was rule-based, but it requires creating strict rules, if the form of the using term is changed, these rules may not work [8].

In this example, the expression "180\ \ 110mm" has a different separator than the usual "180/110 mm", and for it to be correctly extracted, there must be a set of rules describing all cases when the words are used in the context

C. Vital signs (features)

Vital signs are measurements of the body's most basic functions. The five main vital signs routinely monitored by medical professionals and health care providers include the following:

- Blood pressure
- Heart rate
- Respiration rate (rate of breathing)
- Height
- Weight

Vital signs are useful in detecting or monitoring medical problems. Vital signs can be measured in a medical setting, at home, at the site of a medical emergency, or elsewhere. This information also might be extracting from electronic health records, usually as non-structure medical text.

D. Method

Dependency parsing might be used as a component for solving many text mining problems. It is often used as a feature to train machine learning algorithms or used for rule-based approaches for relation extraction [9]. However, as continued research and community efforts in the form of the CoNLL shared tasks, for example, show, dependency parsing in general is still not a problem solved completely [10]. Making a fast, reliable dependency parser readily available for the wider research community for further processing to build upon will help spur efforts in event and relation extraction. As previous research has shown, providing named entity information to the dependency parser can improve the accuracy of the parses. The reasoning is that dedicated named entity recognition (NER) tools perform much better in their specific domain, and by extracting named entities with higher accuracy will facilitate appropriate parsing tools that are not trained on biomedical data.

We used an open-source natural language processing (NLP) library «SpaCy». It is written in Python that performs tokenization, Part-of-Speech (PoS) tagging and dependency parsing. It is the fastest NLP parser available, and offers state-of-the-art accuracy [13].

The most recent extensive evaluation of existing dependency parsers has been performed by Choi et al. [12]. They evaluate 10 different off-the-shelf parsers for accuracy and speed; reporting labeled attachment scores (LAS) of 85% to 90%. While spaCy does not perform the most accurate in their evaluation, it performs fastest maintaining comparable accuracy. SpaCy’s models are statistical and every “decision” they make – for example, which part-of-speech tag to assign, or whether a word is a named entity – is a prediction. This prediction is based on the examples the model has seen during training. To train a model, you first need training data – examples of text, and the labels you want the model to predict. This could be a part-of-speech tag, a named entity or any other information.

The SpaCy library does not offer a pretrained model for the Russian language, but provides an opportunity to conduct training and get a model on its own. SpaCy 2. 0 offers new neural models for tagging, parsing, and entity recognition.

The spaCy package provides the following modules for training and model creation.

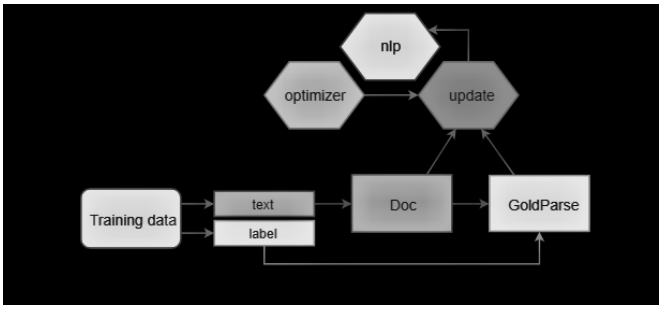

Fig. 2. The process of training model [13]

Our models were designed and implemented from scratch specifically for SpaCy to give you an unrivaled balance of speed, size, and accuracy. Bloom's new embedding strategy with word features is used to support huge dictionaries in tiny tables. Convolutional layers with residual links are used.

The parser and NER use an imitation learning objective to deliver accuracy in-line with the latest research systems, even when evaluated from raw text. This convolutional layer is shared between the tagger, parser and NER, and will also be shared by the future neural lemmatizer. SpaCy v2.1 introduces a new CLI command, SpaCy pretrain, that can make your models much more accurate. It's especially useful when you have limited training data. The spacy pretrain command lets you use transfer learning to initialize your models with information from raw text, using a language model objective similar to the one used in Google's BERT system.

When training a model, we don't just want it to memorize our examples – we want it to come up with a theory that can be generalized across other examples [13]. After all, we don't just want the model to learn that this one instance of "рост" right here is a "increase, growing" – we want it to learn that "рост", in other contexts, is most likely a "height". That's why the training data should always be representative of the data we want to process. This also means that to know how the model is performing, and whether it's learning the right things, you don't only need training data – you'll also need evaluation data. If you only test the model with the data it was trained on, you'll have no idea how well it's generalizing. If you want to train a model from scratch, you usually need at least a few hundred examples for both training and evaluation. To update an existing model, you can already achieve decent results with very few examples – if they're representative.

Example of data description for training from the program in Python:

```
TRAIN_DATA = [
        ("Google rebrands its business apps",
{"entities": [(0, 6, "ORG")]}),
        ("Рост 160 см вес 60 кг состояние
удовлетворительное.",{"entities": [(0, 11,
'HEIGHT'), (12, 21, 'WEIGHT')]})
            ]
```

Example of data description for training from the command line interface (CLI):

```
[
 {"id": 0, "paragraphs":
  [
   {"raw": "Рост 160 см вес 60 кг  состояние
удовлетворительное.","sentences":
    [
    {"tokens":
     [
      {"id": 0, "orth": "Рост","ner": "B-РОСТ" },
      {"id": 1,"orth": "160", "ner": "I-РОСТ"},
      {"id": 2, "orth": "см", "ner": "L-РОСТ" },
      {"id": 3,"orth": "вес","ner": "B-ВЕС"},
      {"id": 4, "orth": "60","ner": "I-ВЕС"},
      {"id": 5, "orth": "кг","ner": "L-ВЕС"},
      {"id": 6, "orth": "состояние""ner": "O"},
      { "id":7,"orth": "удовлетворительное","ner":
"O"},
      {"id": 8, "orth": ".","ner": "O"}
     ],
     "brackets": []
    }
   ],
  "cats": []
  }
  ]
 }
]
```

E. Difficulty for training

An important problem with text processing for feature extraction is the large number of errors and typos. Due to the fact that the electronic medical history is an official medical document signed by a doctor, we do not have the right to make any corrections and adjustments. In this case, we have to do preprocessing of the input text "on the fly", correcting errors, typos and abbreviations.

SpaCy provide the ways to increase the accuracy of new model, when you create the model from scratches. The models take a very long time to train, so we can't run enough experiments to figure out what's the best hyper parameters for train our model. We use the following algorithms for optimization:

Initialize with batch size 1, and compound to a maximum determined by your data size and problem type.

- Use Adam solver with fixed learning rate.
- Use averaged parameters
- Use L2 regularization.
- Clip gradients by L2 norm to 1.

On small data sizes, start at a high dropout rate, with linear decay. We developed this experimentally.

F. Results

The model was trained for 120 epochs, using SGD optimizer and binary cross entropy as the loss function. The loss function results were also weighted according to the proportion of samples with positive and negative samples. The model was evaluated using F1 score, precision and recall. The F1 scores for each label were average weighted by support. We train our model using the spacy train command.

The Scorer and nlp.evaluate now report the text classification scores, calculated as the F-score on positive label for binary exclusive tasks, the macro-averaged F-score for all

exclusive features. Most tokens in real-world medical documents are not part of entity names as usually defined, so the baseline precision, recall and F1 is extravagantly high, typically >90%; going by this logic, the entity wise precision recall values are reasonably good.

As a result of the training, we obtained the following metrics for our model (Table I).

TABLE I. MODEL METRICS FOR TRAINING

Feature	NER Precision	NER recall	NER F-score
BP (blood pressure)	99.8	99.7	99.8
WEIGHT	99.7	99.4	99.6
HEIGHT	98.5	100	99.3
HR (heart rate)	98.4	98.2	98.7
BR (breeze rate)	100	100	100
MODEL	99.4	99.6	99.5

III. CONCLUSION

The model we obtained showed good results in extracting features from unstructured disease histories and EHR. The model practically does not require large computer resources and can be integrated into any medical information system. At the moment, the model is embedded in the CDSS "Webiomed" (webiomed.ai) and is undergoing comprehensive testing.

The model we obtained showed good results in extracting features from unstructured EHR. The model practically does not require large computer resources and can be integrated into any medical information system. At the moment, the model is embedded in the clinical decision support system (CDSS) "Webiomed" [14] and is undergoing comprehensive testing.

This technique allows to use a very large amount of data from patient's electronical health records, we are talking about hundreds of thousands, which is almost impossible in clinical studies, such as Framingham Heart Study, where about 10 thousand patients were examined [15]. Some features of the system may be available for use by patients, such as viewing the EHR. [16]

REFERENCES

[1] Angus Roberts, "Language, Structure, and Reuse in the Electronic Health Record", *Ama Journal of Ethics*, 2017,vol. 19(3), pp. 281-288.

[2] Meystre S, Savova GK, Kipper-Schuler KC, Hurdle JF, "Extracting information from textual documents in the electronic health record: a review of recent research", *Yearb Med Inform.,* 2008, pp.128-144.

[3] Boyang Zhao, Clinical Data Extraction and Normalization of Cyrillic Electronic Health Records Via Deep-Learning Natural Language Processing. JCO Clinical Cancer Informatics , published online October 2, 2019. DOI:10.1200/CCI.19.00057

[4] Andersson, C., Johnson, A.D., Benjamin, E.J. et al. 70-year legacy of the Framingham Heart Study. *Nat Rev Cardiol* 16, 687–698 (2019). https://doi.org/10.1038/s41569-019-0202-5

[5] Malmasi S, Ge W, Hosomura N, Turchin A. Comparison of Natural Language Processing Techniques in Analysis of Sparse Clinical Data: Insulin Decline by Patients. *AMIA Joint Summits on Translational Science*. 2019 ;2019:610-619Group, 2014.

[6] Sager N, Lyman M, Bucknall C, Nhan N, Tick LJ. Natural language processing and the representation of clinical data. J Am Med Inform Assoc. 1994;1(2):142-160.

[7] I Korsakov, A Gusev, T Kuznetsova, D Gavrilov, R Novitskiy, Deep and machine learning models to improve risk prediction of cardiovascular disease using data extraction from electronic health records, *European Heart Journal*, Volume 40, Issue Supplement_1, October 2019, ehz748.0670, https://doi.org/10.1093/eurheartj/ehz748.0670

[8] L. Sathish Kumar1 and A. Padmapriya, Rule Based Information Extraction from Electronic Health Records by Forward-Chaining, Article in Elsevier Ergonomics Book Series · August 2014

[9] Colic N. Dependency parsing for relation extraction in biomedical literature [Master Thesis in Computer Science]. *Zurich: University of Zurich*, 2016, Immatriculation Number: 09-716-572.

[10] Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C. Neural architectures for named entity recognition. Ithaca: arXiv, Cornell University, 2016. Accessed 2019 Apr 2. Available from: https://arxiv.org/abs/1603.01360.

[11] Honnibal M, Johnson M. An improved non-monotonic transition system for dependency parsing. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015 Sep 17-21, Lisbon, Portugal. Stroudsburg: Association for Computational Linguistics, 2015. pp. 1373-1378

[12] Jinho D. Choi, Joel R. Tetreault, Amanda Stent. "It Depends: Dependency Parser Comparison Using A Web-based Evaluation Tool", Conference: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics At: July 2015.Beijing, China

[13] Training spaCy's Statistical Models, Web: https://spacy.io/usage/training

[14] Clinical decision support system Webiomed". Web: https://webiomed.ai

[15] Ross MK, Wei W, Ohno-Machado L, . "Big data" and the electronichealth record. *Yearb Med Inform*. 2014;9(1), pp. 97-104.

[16] Kind EA, Fowles JB, Craft CE, Kind AC, Richter SA. "No change in physician dictation patterns when visit notes are made available online for patients". *Mayo Clin Proc*. 2011;86(5), pp. 397-405.