Experiencer detection and automated extraction of a family disease tree from medical texts in Russian language

Ksenia Balabaeva[1] and Sergey Kovalchuk[1]

[1] ITMO University, Saint-Petersburg, Russia
kyubalabaeva@gmail.com
sergey.v.kovalchuk@gmail.com

Abstract. Text descriptions in natural language are an essential part of electronic health records (EHRs). Such descriptions usually contain facts about patient's life, events, diseases and other relevant information. Sometimes it may also include facts about their family members. In order to find the facts about the right person (experiencer) and convert the unstructured medical text into structured information, we developed a module of experiencer detection. We compared different vector representations and machine learning models to get the highest quality of 0.96 f-score for binary classification and 0.93 f-score for multi-classification. Additionally, we present the results plotting the family disease tree.

Keywords: EHRs, natural language processing, family disease tree, word embeddings, text classification, language models, information retrieval, family history

1 Introduction

In Russia electronic health records (EHRs) consist of structured, half-structured and unstructured text data. In order to process texts and get valuable information, medical experts spend a lot of time on reading. Although, in the era of automatization and big data analysis, researchers and engineers have developed automated tools for such tasks [1, 2]. Even though in most cases text processing is a relatively simple task when it comes to structured or half-structured texts, unstructured text mining is still a challenge. It requires more intelligent and computationally complex methods for natural language processing.

This work in broad terms is dedicated to unstructured medical text processing and information retrieval. One of the widely used parts of EHRs free text is a patient's life and medical anamnesis that describes facts, events, and diseases. However, there is also information about relatives and hereditary diseases, which a patient may not have. In order to segment text parts describing family members, we have developed a module of binary sentence classification. On the one hand, it filters non-patient information for further processing, while on the other hand it helps to get structured relatives features

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_45

that can be used in further analysis and modelling. The second block of our module includes retrieval of family disease tree from texts in natural language. We have developed a multiclassification algorithm that identifies class of the relatives and maps their information to the family disease tree. To clarify the task setting, the input and output data on each step are displayed on Figure 1.

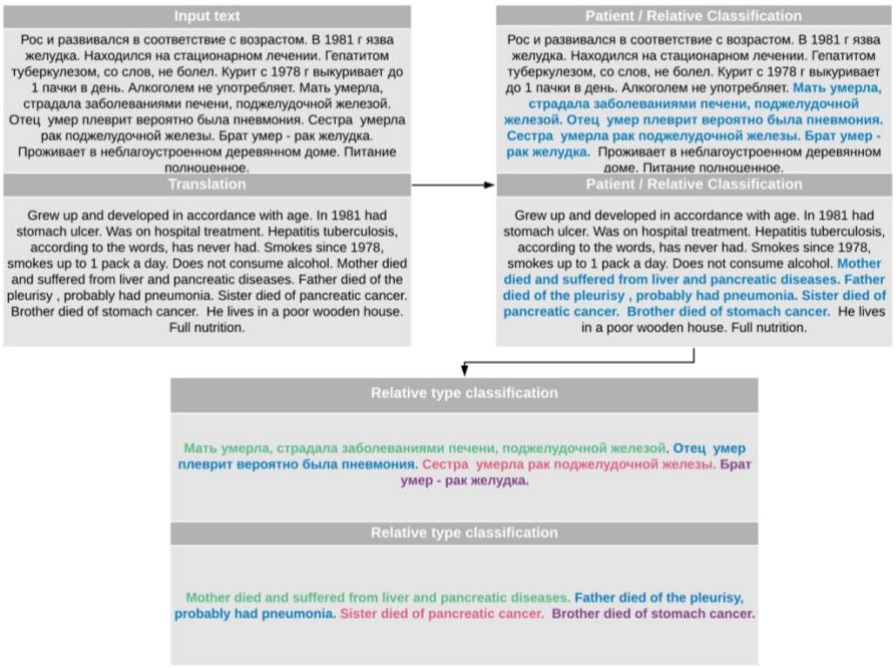

Fig. 1 Main steps of text processing in experiencer detection module. Grammar and spelling of the original text are preserved

It is also necessary to mention that we worked with texts in Russian language, therefore, our work also contributes to the research of languages with complex systems.

2 Literature Review

There are several works in the field of medical 'free text' processing, such as keyword extraction, and disease classification [3, 4]. Examples of similar task to the family disease tree extraction are also featured in literature, for instance, in [5], researchers extract family history from clinical notes, based on the word-wise labeled dataset and applying machine learning (ML) methods. Another work uses algorithms based on syntax structure of the English language [6]. One of the latest works [7] applies deep

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_45

learning algorithms to extract family history. However, they are all applied to English language texts. While some modules of retrieving family tree structure from text are based on rules [8], alt, they work only with well-structured texts not observing medical context.

In general, the task of sentence classification is not new. It has several specific features. First of all, there is a problem of word representation: the length of the sentence varies, but the feature vector must be fixed. There is a classical approach for this problem, bag of words (BOW) vector representations [9], when sentences are represented as collections of words using their text frequency. However, it is a simplification of real interactions between words since BOW disregards the order.

There is also a big contribution to the word representation task from deep learning which is used to get continuous bag of words representation (CBOW) [10]. Another approach is to use skip grams (SG) that enrich representations with word's n-grams [11].

There are also end-to-end solutions in deep learning. For instance, such architectures as convolutional neural networks, LSTM and attention-based models are widely used in sentence classification [12,13,14].

However, the efficient application of deep learning methods usually require big volumes of data. Moreover, the majority of works uses English language for modelling, and the efficiency of the aforementioned methods on other languages is not fully researched, especially for the Russian language.

Another important issue of the family disease tree extraction is the relation bond uncertainty. For example, it is not always possible to say to which exact parent a grandmother belongs to or whether a patient's brother is a half-brother or a sibling. That is why, our goal is to map texts to the risk probability model, concerning the uncertainty in data, uncertainty in a model and uncertainty of the disease inheritance process.

3 Previous work

The module of experiencer detection is a part of our research group text mining project (Figure 2). It is dedicated to the medical language corpora with a high share of specific terminology. The project consists of several functional parts. The first part solves the problem of misspelling correction [15]. The second module is a negation detector, that helps to deal with rejections of events [16]. The third is the experiencer detection module, described in this paper.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_45

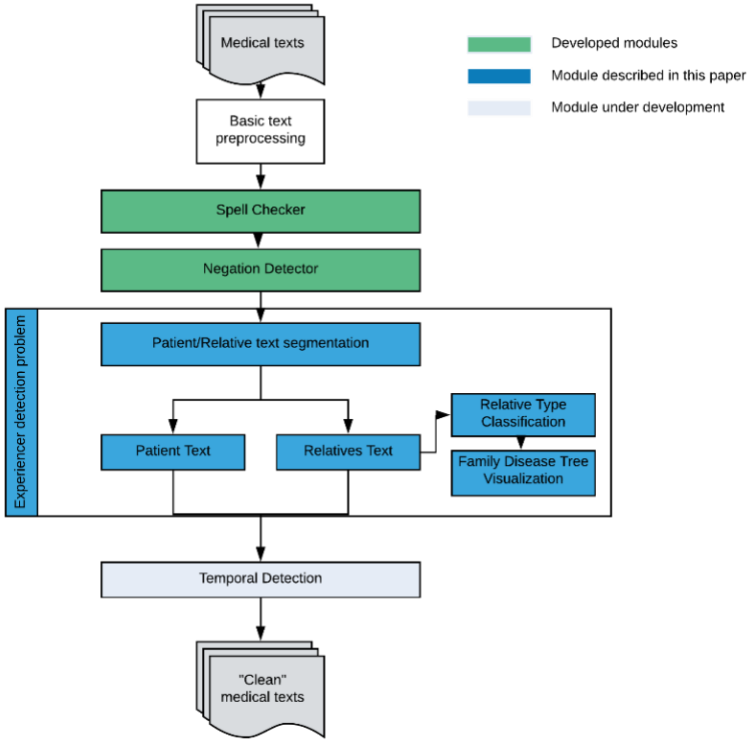

Fig. 2 Experiencer detection module as a part of medical text mining project

The module of temporal structure processing, that helps to assign time stamps to the facts in a sentence is still under research. The green blocks represent the modules that are already developed. The blue ones are currently under development. As a result, all modules will be implemented as a Python 3 package.

4 Data

We use a dataset of electronic health records (EHRs) from Almazov National Medical Research Centre, Saint Petersburg Russia. For the task of patient/non-patient classification, we selected 1376 sentences from 696 patients, with 672 sentences containing information about relatives and 704 sentences describing patients. All sentences were labeled by the authors.

For the second task, the number of unique labels was much higher, and 672 sentences were not enough. Therefore, we decided to enrich our dataset and labeled additional data. Finally, we got 1204 sentences. The labels included 9 most common

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_45

classes: mother, father, sister, brother, daughter, son, grandmother, grandfather, aunt, uncle. The distribution of classes is displayed on Figure 3.

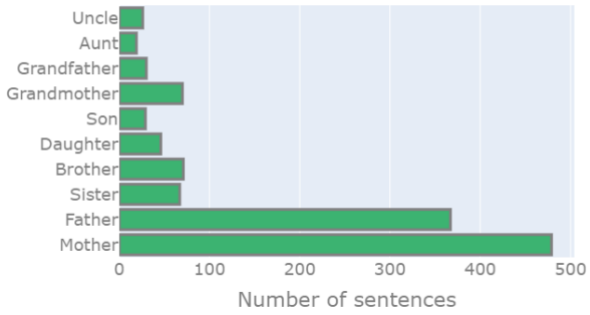

Fig. 3 Relatives' types distribution

The information about parents and their diseases in anamnesis is the most frequent. Sisters, brothers and grandmothers are discussed with approximately equal frequency. Less attention is payed to children, aunts and uncles' diseases.

We also have to mention the restrictions we implied on the data used in the paper. First of all, we only used sentences with one relative, as most sentences (78%) in fact contained information about only one relative. Secondly, multiple subjects in a sentence require more time-consuming labeling process, different task setting and a syntax trees parser. Unfortunately, we have not found satisfying syntax tree parser for Russian language yet. However, we have plans to work on that problem in future.

5 Methods

We have decomposed the module into two separate tasks: patient/relative binary classification and patient type multiclassification. All the experiments were implemented in Python programming language, version 3.7. We started our experiments with baseline development. For each task, an algorithm based on key words search was developed. As key words we used stems of common words for the names of family members. For instance, 'бабушк' for the word 'бабушка' (grandmother) or 'матер', 'мам' for the word 'матери', 'мама' (mother). We used lemmas to find different forms of words regardless of case and number and assigned the class to the sentence according to the key word found in it. This procedure reflects the most common and easiest way to get features from the initial medical texts, although, it is not accurate enough. More details on performance are provided in the results section.

The whole natural language text preprocessing was reduced to removal of punctuation and misspelling correction with the spell checker module already developed by our team [15, 16]. We deliberately avoided normalization and stop words

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_45

removal, since anamnesis contain a lot of specific terms that cannot be normalized correctly using the existing open-source libraries. The reason to keep stop words lies in complex wording in Russian where they can be helpful.

After preprocessing, the text was separated into sentences and vectorized. For vectorization we applied and compared several approaches: bag of words (BOW), continuous bag of words (CBOW) and skip-grams (SG). BOW is a model that represents each sentence as a bag or set of words, disregarding order and grammar. It provides sparse vectors with the length of vocabulary and calculates the scoring of term frequency. An example of BOW representation is TFIDF vectorizer and CountVectorizer (Python sklearn implementation). While CountVectorizer simply counts word frequencies, TFIDF also uses inverse document frequency. However, bag of words approaches have several drawbacks, such as sparsity and lack of sentence meaning. That is why we decided to compare them with word embeddings that build a language model and learn text features . Word embeddings have a huge advantage of word order and context consideration. Therefore, such methods to a certain extent can catch the meaning of the sentence.
In this paper we compare the continuous bag of words (CBOW) and skip-grams, using the FastText model [9]. The difference between these approaches lies in the learning process. While CBOW aims to predict a word by the context, SG is learning to predict context by a word [10].

After that, vectors are input to the ML algorithms to start supervised learning. As models, we use logistic regression, k-nearest neighbors and random forest (python, sklearn library) and also gradient boosting (python, xgboost library) since they have different structure, suit for both binary and multiclassification, and show good performance in many tasks. The hyperparameters were tuned using GridSearch algorithm.

All models are evaluated using 5-fold cross-validation on the quality and stability. As a metric, f-score is used with macro-averaging.

6 Results

6.1 Binary classification for patient and relative sentences

Concerning the results of sentence classification on patient and relatives classes, the baseline model based on key words search had 0.6337 f -score. Fortunately, other approaches performed with a higher quality (Table 1.). In terms of vectorization, bag of words methods (such as CV, TFIDF), on average, performed almost equal results to the skip-gram approach, achieving ~0.95 f-score. The continuous bag of words models show a much lower performance for all ML models. However, the highest score in most trials was still achieved by the SG word embedding approach (f-score 0.96).

Comparing different ML models, there is little difference in their best performances, and the f-score varies in the range of 0.95 to 0.96, even for simpler models such as

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_45

LogReg or KNN. In terms of stability, cross-validation scores have, on average, standard deviation from one to two hundredth and the scores are stable enough.

Table 1. Task 1: Patient-relative classification. F1-scores on 5-fold cross-validation ± std.

Vectorization Method\ Model	Logistic Regression	XGB	Random Forest	KNN
CountVectorizer (CV)	**0.9592** (±0.0107)	0.9344 (±0.0024)	0.9519 (± 0.0082)	0.9236 (±0.0117)
TfIdf Vectorizer (TFIDF)	0.9460 (±0.0107)	0.9337 (±0.0167)	0.9534 (± 0.0071)	0.9207 (±0.0128)
CBOW	0.7211 (±0.0095)	0.8637 (±0.0300)	0.8542 (± 0.0229)	0.8200 (±0.0270)
SG	0.9468 (±0.0292)	**0.9621** (±0.0178)	**0.9563** (±0.0201)	**0.9614** (±0.0151)

6.2 Patient Type Classification

The second task required building a multiclassification algorithm, for 9 types of relatives. The baseline model based on the key word search achieved f-score of 0.59278. The metrics for other approaches are displayed in Table 2. The best vectorization method for this task was bag of words (CountVectorizer). Word embeddings performed worse than even the baseline model. The reason of such results may lay in the size of our dataset. It was only 1204 samples and for several rare classes (Figure 3), such as aunts, uncles or sons contained only 20-30 examples, which may not be sufficient for learning a language model based on neural networks.

Table 2. Task 2: Relative type multi-classification. F1-scores on 5-fold cross-validation ± std.

Vectorization Method\ Model	Log Reg	XGBoost	Random Forest	KNN
CountVectorizer	**0.9235** (±0.0181)	**0.9305** (±0.0285)	0.9123 (± 0.0189)	**0.6895** (±0.0274)
TfIdf Vectorizer	0.7107 (±0.0369)	0.9297 (±0.0324)	**0.9145** (± 0.0145)	0.6207 (±0.0458)
CBOW	0.0914 (±0.0056)	0.2054 (±0.0317)	0.1607 (± 0.0174)	0.1200 (±0.0150)
SG	0.2203 (±0.006)	0.4781 (±0.0509)	0.4236 (±0.04327)	0.38754 (±0.0426)

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_45

Unlike the previous task, not all ML model performed equally good this time. The worst results (0.6895) was shown by K-nearest neighbors algorithm. The best performance was demonstrated by the XGBoost, that achieved 0.93 f-score. Logistic Regression and Random forest also showed quite close and accurate results.

In general, current results are satisfactory, . On the plot (Fig. 4) the results of XGBoost predictions made on vectors from count vectorizer are presented on the left and skip-grams are on the right. The count vectorizer representation provides almost 100% accuracy for the majority of relatives. However, it struggles to correctly predict sentences about brother, son and uncle. It usually confuses them with mother or father which have a higher share in sentence samples (Fig. 3.). Thus, according to the confusion matrix for the skip gram vectorization, the performance of the embeddings strongly depends on the sample size. It is clear, that the quality of predictions is much higher for sentences about mother and father and other classes suffer from the lack of examples. In order to improve the embedding results, the data may be enriched with more instances for the other types of family members and make the dataset more balanced. According to the results, at least 3-5 hundreds of samples for each relative type are required.

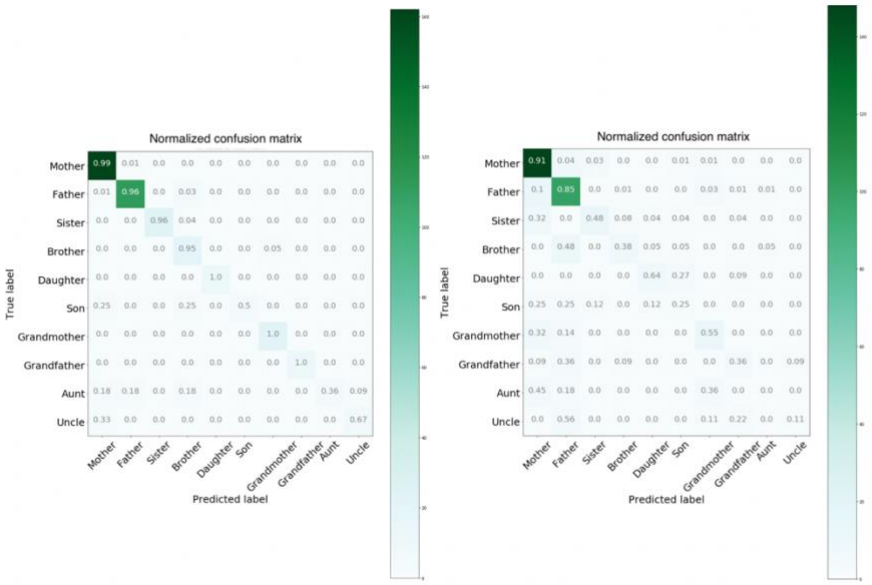

Figure 4 Confusion Matrix for predictions of XGBoost + CountVectorizer (on the left) and Skip-Gram (on the right)

In addition to the model's prediction, we have implemented visualization of family disease tree as an output of the module (Fig. 5). As an input it takes sentences about relatives with the assigned relative type from the predictive models.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_45

Figure 5. Family disease tree visualization

Then, it extracts the disease or events that a relative experienced with simple syntax rules (for instance, removal of verbs and words, and a subject from the sentence). After that it plots a tree-type graph using Plotly Python library.

7 Conclusion

In conclusion, set goal is achieved and the module of experiencer detection for medical texts is developed. To reach this goal, the literature was analyzed, a dataset was collected and labeled, and experiments on the methods of vectorization and ML models were performed. A model that scored 0.96 f-metrics for patient/relative classification and 0.93 f-score on relative type classification problem was built. It was also shown that naïve methods based on the key words search can hardly solve the task and the developed models perform better.

The findings can be used for hereditary disease modelling. The module may be applied not only to medicine, but also to other spheres, where it is required to determine the subject in a sentence, for instance, in social networks or advertisements. Moreover, this approach can be applied to other languages, that do not have a language corpus or a syntax tree parser.

However, there are still things to improve. For instance, we need to expand the functionality to process the sentences with multiple subjects, for example, when there is information concerning several relatives in one sentence. For this task, the dataset should be labelled on the word level and the syntax-tree parser is to be implemented. Moreover, we need to expand the dataset especially for the task of relative type classification and try to generate and use synthetic data.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_45

As this module is a part of the medical text mining project, infrastructure for communication with other modules should be provided. As a result, a python 3 package will be developed in future.

Acknowledgement

This work is financially supported by National Center for Cognitive Research of ITMO University.

References

1. Hanauer, D.: Supporting Information Retrieval from Electronic Health Records: A Report of University of Michigan's Nine-Year Experience in Developing and Using the Electronic Medical Record Search Engine (EMERSE). J Biomed Inform 55, 290-300 (2015)
2. Cesar dos Reis, J., Perciani, E.: Intention-Based Information Retrieval of Electronic Health Records. 25th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprise. (2016)
3. Tang, M., Gandhi, P., Kabir, M.: Progress Notes Classification and Keyword Extraction using Attention based Deep Learning Models with BERT. Preprint, https://arxiv.org/pdf/1910.05786.pdf, last accessed 2020/02/07, last accessed 2020/02/07.
4. Zhang, X., Henao, R., Gan, Z., Multi-Label Learning from Medical Plain Text with Convolutional Residual Models. Preprint, https://arxiv.org/pdf/1801.05062.pdf, last accessed 2020/02/07.
5. Bill, R., Pakhomov, S., Chen, E.,: Automated Extraction of Family History Information from Clinical Notes. AMIA Annu Symp Proc. Pp.1709-1717, (2014).
6. Azab, M., Dadian, S., Nastase, V.,: Towards Extracting Medical Family History from Natural Language Interactions: A New Dataset and Baselines. EMNLP/IJCNLP, (2019).
7. Lewis, N., Gruhl, D., Yang, H.,: Extracting Family History Diagnoses From Clinical Texts, BICoB (2011).
8. Family tree maker, github project: https://github.com/adrienverge/familytreemaker/blob/master/familytreemaker.py, last accessed 2020/02/07.
9. Zhang, Y., Jin, R., Zhou, Z.,: Understanding bag-of-words model: A statistical framework. International Journal of Machine Learning and Cybernetics 1(11), 43-52 (2010).
10. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. CONFERENCE 2013, Proceedings of the International Conference on Learning Representations, ICLR, (2013).
11. Mikolov, T., Sutskever, I., Chen, K., Corado, J., Dean, J.,: Distributed representations of words and phrases and their compositionality. NIPS'13: Proceedings of the 26th International Conference on Neural Information Processing Systems, vol.2, pp.3111-3119, (2013).
12. Kim, Y.,: Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746-1751, (2014).
13. Ganda, R., Mahmood, A., Deep learning for sentence classification, Conference: 2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT), (2017).

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_45

14. Zhou, Q., Wang, X., Differentiated Attentive Representation Learning for Sentence Classification. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), (2018).
15. Balabaeva, K., Funkner, A., Kovalchuk, S.,: Automated Spelling Correction for Clinical Text Mining in Russian. Preprint, https://arxiv.org/abs/2004.04987 , last accessed 2020/04/15 (2020).
16. Funkner, A., Balabaeva, K., Kovalchuk, S.,: Automated negation detection for medical texts in Russian Language. Preprint, https://arxiv.org/pdf/2004.04980.pdf , last accessed 2020/04/15 (2020).

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_45