DocKG: A Knowledge Graph Framework for Health with Doctor-in-the-Loop

Ming Sheng[1], Jingwen Wang[2], Yong Zhang[1(✉)], Xin Li[3], Chao Li[1], Chunxiao Xing[1], Qiang Li[4], Yuyao Shao[2], and Han Zhang[5]

[1] RIIT&BNRCIST&DCST, Tsinghua University, Beijing 100084, China
{shengming, zhangyong05, li-chao, xingcx}@tsinghua.edu.cn
[2] Beijing Foreign Studies University, Beijing 100089, China
{wjwen, shaoyuyao}@bfsu.edu.cn
[3] Beijing Tsinghua Changgung Hospital Medical Center, Tsinghua University, Beijing, China
horsebackdancing@sina.com
[4] Center for Science and Technology Talents, MoST, Beijing 100045, China
liq@sttc.net.cn
[5] Beijing University of Posts and Telecommunications, Beijing 100876, China
zhanghan3281@bupt.edu.cn

Abstract. Knowledge graphs can support different types of services and are a valuable source. Automatic methods have been widely used in many domains to construct the knowledge graphs. However, it is more complex and difficult in the medical domain. There are three reasons: (1) the complex and obscure nature of medical concepts and relations, (2) inconsistent standards and (3) heterogeneous multi-source medical data with low quality like EMRs (Electronic Medical Records). Therefore, the quality of knowledge requires a lot of manual efforts from experts in the process. In this paper, we introduce an overall framework called DocKG that provides insights on where and when to import manual efforts in the process to construct a health knowledge graph. In DocKG, four tools are provided to facilitate the doctors' contribution, i.e. matching synonym, discovering and editing new concepts, annotating concepts and relations, together with establishing rule base. The application for cardiovascular diseases demonstrates that DocKG could improve the accuracy and efficiency of medical knowledge graph construction.

Keywords: Medical knowledge graph construction · Doctor-in-the-loop · EMR

1 Introduction

Knowledge graph serves as a repository that can integrate information from different sources together. There are concept knowledge graphs which contain only the concepts, and instance graphs which contains instance from the real world. In the medical domain, knowledge graph can support services like disease prediction, medication recommendation [1], etc. The quality of the services depends on the quality of the

H. Wang et al. (Eds.): HIS 2019, LNCS 11837, pp. 3–14, 2019.
https://doi.org/10.1007/978-3-030-32962-4_1

knowledge. Many attempts have been made to build the knowledge graphs completely automatically [2–4] from the data on the Internet [5]. When it comes to medical knowledge graphs, these fully automatic methods seem to be inadequate because of the following reasons:

1. The concepts and relations in medical domain are complex and obscure.
2. The data in medical domain follows inconsistent standards.
3. The data in medical domain is from different sources, heterogeneous and with poor quality.

Therefore, the general ways that have been used to construct knowledge graph automatically cannot be directly applied to the medical domain. On the one hand, it's very necessary to include some experts' experience in the process to improve the quality. On the other hand, if too many human activities are involved in the process, tremendous amount of expert time and effort will be needed and the efficiency of the whole construction progress will be too low [6]. What's worse, the whole system will be too brittle and unable to adapt to or expand to other new medical topics [7]. Therefore, an automatic method that involves an appropriate amount of expert effort is required. Therefore, the balance between the experts' effort and the knowledge graph construction is very delicate and needs to be carefully examined.

In this paper, we introduce DocKG, an overall framework that sheds light on when and where expert effort, also known as doctor-in-the-loop mechanism is needed to improve both the efficiency and quality of medical knowledge graph construction.

This paper is organized as follows. In Sect. 2, previous work related to DocKG will be introduced. In Sect. 3, the overall framework and workflow will be presented. In Sect. 4, when and where to involve doctors, i.e. doctor-in-the-loop will be explained in detail. Section 5 will summarize the paper and discuss possible improvements in the future.

2 Related Work

In this section, we will compare several mainstream knowledge graph building tools, and investigate human-in-the-loop mechanism.

2.1 Knowledge Graph Building Tools

Many automatic knowledge graph building tools have been proposed to process massive data and construct knowledge graphs without human involvement. Below is a summary of some typical knowledge graph building tools:

As Table 1 shows, the mainstream knowledge graph building tools include RDR, cTAKES, pMineR, I-KAT, etc. Among the six tools, less than half of them involve human activities in the construction process. None of them contain all of the four common functions in one single tool. Therefore, it's very inconvenient for the doctors and data engineers to build a high-quality medical knowledge graph with these tools.

Table 1. Comparison of functions between different building tools.

Name	Field	Data source	Entity recognition	Relation extraction	Entity alignment	ER/RDF mapping	Expert Involve
RDR [8]	Medical	–	×	×	×	×	√
cTAKES [9]	Medical	UMLS	√	√	√	×	×
pMineR [10]	Medical	EMR	×	×	×	×	×
I-KAT [11]	Medical	SNOMED-CT	×	×	×	√	√
myDIG [12]	General	csv, JSON	√	√	×	×	×
semTK [13]	General	csv…	×	×	×	√	×
DocKG	Medical	UMLS, EMR…	√	√	√	√	√

2.2 Human-in-the-Loop

In the medical domain, automatic methods based on machine learning have achieved promising results in many aspects like disease prediction and clinical notes classification. Despite that fact that automatic Machine Learning (aML) in medical domain has drawn many researchers' interest and have been growing rapidly, however, one disadvantage lies in their inexplicability [14] because the internal principles are beyond human's comprehension [15]. What's more, aML requires plenty of training sets to achieve promising results, but in the medical domain the researchers are sometimes faced with a small number of datasets or rare events. Therefore, it is necessary to develop algorithms that can interact with agents (like doctors) and can optimize their learning behavior. Through this interaction, training samples can be selected heuristically, and research time can be reduced dramatically. Algorithms that involves humans' interaction can be defined as "human-in-the-loop" [16]. Human-in-the loop has actually been applied to many aspects of artificial intelligence like named entity recognition [17] and rules learning [15] to improve the performance. However, in medical domain, few attempts have been made to incorporate human-in-the-loop mechanism to improve the performance, especially with regards of knowledge graph construction.

3 Framework and Data Flow

In this section, we are going to present the overall framework of the whole system. In order to explain the system in detail, the workflow of the whole construction process will also be presented. In order to store the information required and generated by DocKG, we used Apache Jena database.

3.1 Framework

In this part, the overall framework and workflow will be introduced based on an existing knowledge graph platform, HKGB.

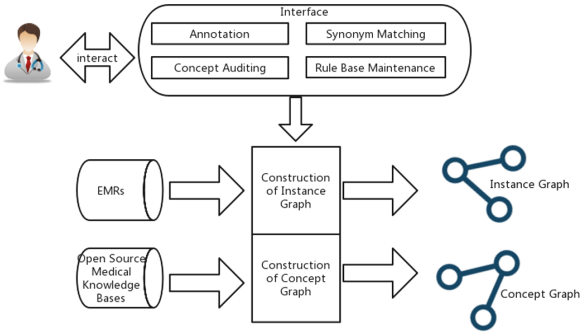

Fig. 1. Framework of medical knowledge graph construction with doctor-in-the-loop.

As Fig. 1 shows, a system that can construct disease-specific medical knowledge graph should include the following parts:

1. Doctors that focus on a particular field of disease.
2. A set of interfaces where the doctors can interact with the whole system.
3. Data sources.
4. Concept graph and instance graph constructors.

In the system, the doctors should be able to interact with the construction process through a set of interfaces. In this way, the doctors can influence the process by "injecting" their experience into the system. In the meantime, a set of automatic tools for constructing a medical knowledge graph is needed. Thus, by providing the interface we managed to combine the doctors' knowledge together with the automatic construction methods.

3.2 Workflow

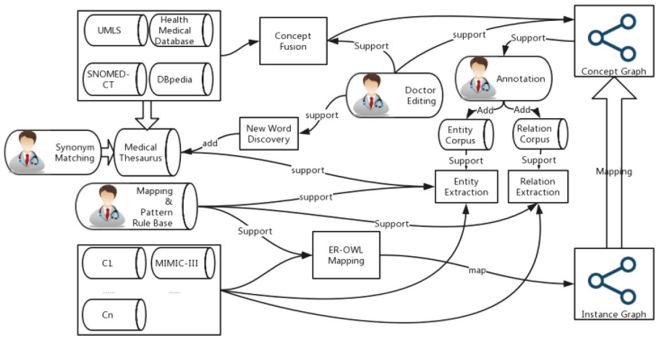

Fig. 2. Workflow of medical knowledge graph construction with doctor-in-the-loop

Figure 2 shows the workflow of the system in detail. In the system, there are four points in which the doctors should be involved.

1. Synonym matching in the fusion and alignment of concepts from different knowledge bases.
2. Doctor editing in the new word discovery and new concept selection.
3. Doctor annotation in the entity and relation extraction from EMRs.
4. Establishment of a rule base that contain both mapping rules and patterns for entity & relation extraction.

The four points will be explained in detail in Sect. 4.

4 Doctor-in-the-Loop in the Construction

In this section, we are mainly going to discuss the four points where the doctors should be involved in detail. Generally speaking, the doctors should provide their opinions on matching synonym, discovering and editing new concepts, annotating concepts and relations, together with establishing rule base to improve the quality of medical knowledge graph. In this section, we will use the construction of a health knowledge graph for cardiovascular diseases as example.

4.1 Synonym Matching Module

Existing medical knowledge bases are very important source of knowledge graph. In order for the information to be fully utilized, the different concepts and relations with the same meaning have to be properly aligned together. To improve the accuracy of the automatic matching methods and efficiency of manual alignment methods, we propose a synonym module that incorporate the results from doctors and matchers. There are two phases in this module: matching phase and aggregating phase (Fig. 3).

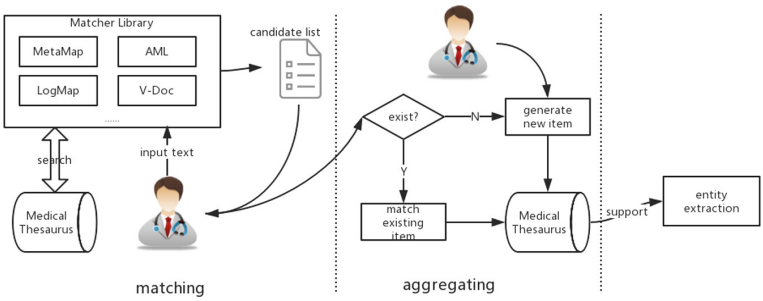

Fig. 3. Workflow of the synonym matching module

This module works on the corpus level and can operate across different data sources. The doctors can input new words or phrases into the module. Then the input text is passed onto the matcher library (a set of different matchers) to be processed. According to the matcher library, a list of candidates that are possible synonym to the input text is returned to the doctors. The list of candidates contains approximately 10 words or so, greatly narrowing down the doctors' search scope. The doctors can then decide by themselves whether the items in the list are synonyms to the input text or not. If there are items in the list that the doctors believe to be synonymous to the input text, then the doctor can align it to one of the existing items that they believe is the best match. If there are not, the doctors can create a new node to integrate the input text into the corpus. The words stored in the thesaurus can then support entity extraction.

The key part of this module is the organization the words and phrases with different spellings, different data sources but the same meaning. To address this problem, we introduce a hierarchical structure. We assign each different concept (words/phrases that have a unique meaning) with a unique concept identity (CID). A concept may have many expressions but only one expression is preferred. This preferred expression is the default representation for the concept. For the expressions that have the same meaning but different spellings or different data sources, we assign each of these expressions with a unique atom identity (AID). The AIDs are child nodes of the corresponding CID. In Fig. 4, we take "Amaurosis Fugax", a typical symptom for heart attack as an example to demonstrate the hierarchical structure.

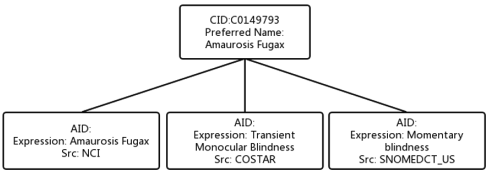

Fig. 4. Hierarchical structure of a concept

4.2 Concept Auditing Module

Unlike the synonym matching module that works on the corpus level, this module works beyond the corpus and on the concept level to provide the doctors with an interface to work directly with the concept graph. This concept auditing module provides two functions to the doctors: (1) concept selection & alignment and (2) new word discovery.

Concept Selection and Alignment
Due to the obscure nature of medical terms, the concepts in the medical thesaurus must be carefully inspected and selected before they can be added to concepts graphs. Figure 5 demonstrates the workflow of concept selection and alignment.

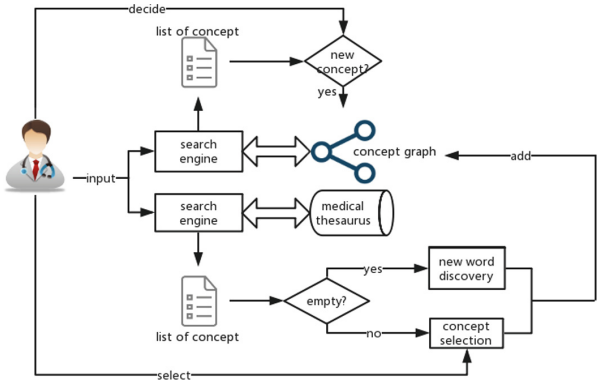

Fig. 5. Workflow of this function

If the doctors want to add a new concept to the concept graph, they can input the text, then the string will be passed onto the search engines on both the medical thesaurus and the conceptual graph. The search engine on the thesaurus will return a list of concepts corresponding to the input strings. The search engine on the conceptual graph will return a list of concepts from the graph that are similar to the concept corresponding to the input string. Instead of having to search manually through the large volume of concepts in the whole thesaurus, the doctors will only need to have a quick scan over the list of concepts provided by the search engines. The doctors can decide by themselves whether the concept corresponding to the input string is a new concept or not. If yes, the doctor can choose one from the list of concepts corresponding to the input string and add to the conceptual graph. However, if the search engine on the medical thesaurus returns no item corresponding to the input string, the doctors should use the new word discovery function described in the following section.

New Word Discovery
The new word discovery function provides the doctors with an interface to customize the terms and concepts that are not in the medical thesaurus. The new terms can be added through the following two methods:

Data-driven Method

This method aims to acquire information from the patients' EMRs. Some features of the EMRs that are not stored in the concept graph can be added.

Table 2. Part of a patient's EMR.

Item	Cardiac apex pulsation	Heart sound A2	Pericardial friction
Result	Accentuated	Split	Normal

Table 2 shows part of a patient's EMR in the cardiology department. Cardiac apex pulsation, heart sound A2 and pericardial friction are all important features for diagnoses on cardiac diseases. However, none of the three features can be aligned with concepts in the concept graph. Under this situation, the doctors can add new concepts to the graph through this module.

Demand-driven Method

The doctors can simply define some concepts and relations based on their own experience. Sometimes the information in the EMRs is simply too complicated and expands over many aspects. The features are too scattered while the doctors only want to narrow their attention down to a few more important features. In this need-driven method, the doctors can leave the EMRs behind and define concepts and relations on a higher level. Figure 6 shows an example of a graph defined by the doctors with a particular focus on the diagnoses of myocardial infarction.

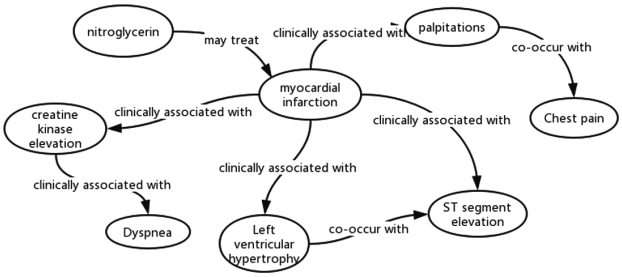

Fig. 6. A high-level graph defined by the doctors

4.3 Entity and Relation Annotation Module

In order to obtain information from the patients' EMRs, entity & relation extraction is needed. The quality of extraction is largely depended on the annotation. However, in the medical domain, there are plenty of entity classes that do not fit the traditionally defined four-class paradigm (PER, LOC, ORG, MISC). For example, in the clinical notes for cardiovascular diseases, there are chest pain location, onset period, etc. If these domain-specific labels are ignored, the quality of extraction based on deep learning methods will decline. Therefore, the annotation module provides the doctors with an interface to annotate the clinical notes in patients' EMRs.

As Fig. 7 shows, this interface is able to load in the patients EMRs and present the clinical notes to the doctors. On the left of the interface listed some pre-defined entity labels, including disease incentive, radiating location, medication name, etc. Apart from these pre-defined labels, the doctors can also customize their own labels. Then the doctors can select words or phrases and assign them with a proper label, or select pairs of entities from the EMR and assign a relation label (shown in Fig. 8) to the pair of entity. The results can then be added to entity & relation annotation respectively to support extraction. The data engineers should focus on using machine learning models, like CRF and CNN-LSTM for automatic extraction, while the doctors can focus on reviewing the results from the models and generating training materials for the models.

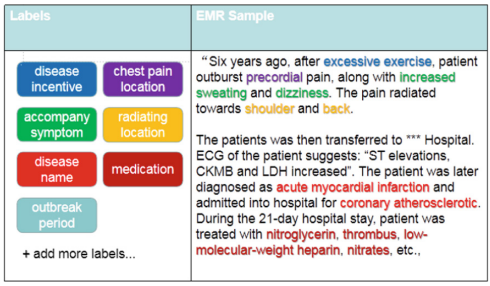

Fig. 7. Interface of entity annotation in detail

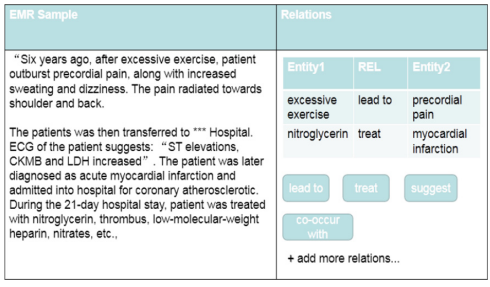

Fig. 8. Interface of relation annotation in detail

4.4 Rule Base Module

In order to support the construction process, there are two types of rules generated by the doctors that need to be stored. One type is mapping rules from ER model to RDF model, another type is extraction rules.

Mapping Rules

The instance graph is described in RDF/OWLS to better present the information in the form of graph. Data stored in ER models needs to be transformed into RDF/OWLS models.

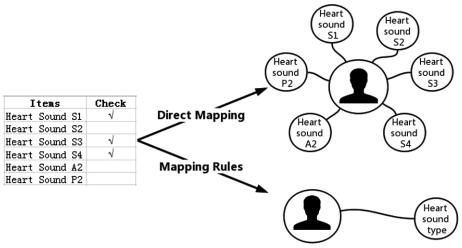

Fig. 9. Mapping process from ER to RDF

As Fig. 9 shows, on the left is an example of the checklist from one cardiovascular patient's EMR. There are six types of heart sounds and the doctors put a mark to suggest that the patient is positive of this symptom. The top right of Fig. 9 shows the direct mapping results: each type of heart sound is assigned to one attribute of the patient. Direct transformation of this ER model may cause the RDF/OWLS to be extremely complex and redundant. However, with the mapping rules (on the bottom right of Fig. 9) defined by doctors, the mapping result can become much simpler and more meaningful. All the six types of heart sound are assigned to one attribute called "heart sound type". With this attribute, the six types of heart sound actually become the values of this one attribute. Table 3 shows an example of mapping rules. The mapping rules will support the construction of instance graph.

Table 3. Example of mapping rules.

ER model	RDF/OWLS model
Heart Sound S1	Heart Sound Type CID:0008123
Heart Sound S2	
Heart Sound S3	
Heart Sound S4	
Heart Sound P2	
Heart Sound A2	
Transfusion of platelet	Blood Transfusion Type CID:0001023
Transfusion of RBC	
Transfusion of plasma	

Extraction Rules

There are two methods for entity extraction: one is based on sequence annotation method and the other is based on rules and patterns. Sequence annotation has been introduced in Sect. 4.3. Here we will focus on rules and patterns for extraction.

Extraction based on rules and patterns has shown some promising results because of its flexibility. This flexibility is especially important in the medical domain because the doctors' demands are frequently changing. By providing the doctors with an interface to customize the rules and patterns, more attention can be put to the more meaningful words that meet the doctors' needs.

First the doctors can define delimiters. Diagnosis of cardiovascular diseases requires extra attention on the patients' symptoms, so the doctors can customize regular expression like "showed symptoms of *". The * can match the words/phrases that suggest symptoms. Then industrial-level NLP tools like spaCy and Jieba with functions of matching and extracting can be applied on the clinical notes. Table 4 shows an example of extraction rules and the extraction result in cardiology. By incorporating the NLP tools, the doctors are freed of the labor to mark the entities manually and can focus more on the information that the text can provide.

Table 4. Example of extraction rules

Input: For further diagnosis and treatment, the patient is admitted to our department. The patient was diagnosed with "unstable angina" and "chest pain" in hospital. Since onset, the patient is conscious, he has been treated with nitroglycerin and heparin after admission.	
PATTERN 1: treated with + * + after admission OUTPUT ENTITY: {nitroglycerin} {heparin}	PATTERN 2: diagnosed with + * + in hospital OUTPUT ENTITY: {unstable angina} {chest pain}

5 Conclusion and Future Work

In the article, we introduce a knowledge graph framework for healthcare based on doctor-in-the-loop. The key point for the construction process is to combine the doctors' efforts with the automatic methods to achieve the balance between accuracy and efficiency. When and where to involve doctors in the loop is emphasized and there are four points in which the doctors' experience needs to be involved. The four points are: (1) synonym matching, (2) new concepts discovering and auditing, (3) concepts and relations annotation, (4) rule base establishment. Therefore, the quality for medical knowledge graph construction could be improved.

Acknowledgments. This work was supported by NSFC (91646202), National Key R&D Program of China (2018YFB1404400, 2018YFB1402700).

References

1. Wu, C., et al.: Prediction of fatty liver disease using machine learning algorithms. In: Computer Methods and Programs in Biomedicine, vol. 170, pp. 23–29 (2019)
2. Martínez Rodríguez, J.-L., López Arévalo, I., Rios Alvarado, A.B.: OpenIE-based approach for Knowledge Graph construction from text. Expert Syst. Appl. **113**, 339–355 (2018)
3. Wang, C., Ma, X., Chen, J., Chen, J.: Information extraction and knowledge graph construction from geoscience literature. Comput. Geosci. **112**, 112–120 (2018)

4. Qi, C., Song, Q., Zhang, P., Yuan, H.: Cn-MAKG: china meteorology and agriculture knowledge graph construction based on semi-structured data. In: ICIS: IEEE Computer Society, pp. 692–696 (2018)
5. Ye, M.: Text Mining for Building a Biomedical Knowledge Base on Diseases, Risk Factors, and Symptoms. Germany: Max-Planck-Institute for Informatics (2011)
6. Rotmensch, M., Halpern, Y., Tlimat, A., Horng, S., Sontag, D.: Learning a health knowledge graph from electronic medical records. Sci. Rep. **7**(1), 5994 (2017)
7. Chen, P., Lu, Y., Zheng, V.W., Chen, X., Yang, B.: KnowEdu: a system to construct knowledge graph for education. IEEE Access **6**, 31553–31563 (2018)
8. Hyeon, J., Oh, K., Kim, Y.J., Chung, H., Kang, B.H., Choi, H.-J.: Constructing an initial knowledge base for medical domain expert system using induct RDR. In: BigComp: IEEE Computer Society, pp. 408–410 (2016)
9. Savova, G.K., et al.: Mayo clinical text analysis and knowledge extraction system (cTAKES): architecture, component evaluation and applications. J. Am. Med. Inf. Assoc. **17**, 507–513 (2010)
10. Gatta, R., et al.: Generating and Comparing Knowledge Graphs of Medical Processes Using pMineR. In: K-CAP: ACM, pp. 36:1–36:4 (2017)
11. Afzal, M., Hussain, M., Khan, W.A., Ali, T., Lee, S., Kang, B.H.: KnowledgeButton: an evidence adaptive tool for CDSS and clinical research. In: INISTA: IEEE, pp. 273–280 (2014)
12. Kejriwal, M., Szekely, P.: myDIG: personalized illicit domain-specific knowledge discovery with no programming. In: Future Internet, vol. 11, p. 59 (2019). https://doi.org/10.3390/fi11030059
13. semTK. http://semtk.research.ge.com/
14. Amaral, A.D., Angelova, G., Bontcheva, K., Mitkov, R.: Rule-based named entity extraction for ontology population. In: RANLP: RANLP Organising Committee/ACL, pp. 58–62 (2013)
15. Yang, Y., et al.: A study on interaction in human-in-the-loop machine learning for text analytics. In: IUI Workshops: CEUR-WS.org, (CEUR Workshop Proceedings), vol. 2327 (2019)
16. Holzinger, A.: Interactive machine learning for health informatics: when do we need the human-in-the-loop? Brain Inf. **3**(2), 119–131 (2016)
17. da Silva, T.L.C., et al.: Improving named entity recognition using deep learning with human in the loop. In: EDBT: OpenProceedings.org., pp. 594–597 (2019)