Reasoning with Textual Queries: A Case of Medical Text

Damir Juric, Giorgos Stoilos, Szymon Wartak, and Mohammad Khodadadi

Babylon Health, London, SW3 3DD, UK

Abstract. Text understanding and reasoning is a very difficult but highly important problem with many practical applications like chatbots. Babylon Health is building an AI-based symptom checking service offered through a chatbot. Depending on what medical terms appear in the text, nodes in a Probabilistic Graph Model (PGM) need to be activated in order to start the symptom checking process. We developed a Semantic Technologies-based solution where OWL concepts are build from user text (in an attempt to capture its meaning) and then compared with respect to subsumption against the conditions in PGM which are encoded using concepts form a medical KB. We developed a knowledge extraction method as well as a hybrid reasoning algorithm that compares concepts using both logical axioms from the medical KB as well as potentially additional information hidden in their labels. We implemented all our algorithms and conducted an experimental evaluation comparing to a baseline text annotation and an ML-based approach obtaining encouraging results.

1 Introduction

Text understanding and reasoning is at the heart of modern chatbots. Users input text which needs to be interpreted in order to activate the respective background services and accomplish the requested task. Babylon Health offers a chatbot which users can use to check their symptoms. Users input text such as "*I am feeling my head is going to explode since this morning*" and subsequently nodes in a Probabilistic Graph Model (PGM) need to be activated in order to ask the user about more specific questions. The nodes in PGM are annotated with classes from a medical KB constructed in Babylon which integrates well-known medical ontologies like SNOMED, NCI, and more [5]. To accomplish the above task medical terms in user text need to be identified and compared against medical concepts in the PGM. A naive approach would be to annotate user text using some text annotator and then check these concepts in the PGM. However, this approach does not capture potential relations between the medical terms in the user text, e.g., the relations between "sever" with "pain"and "morning" in our running example. To understand meaning in text we designed a knowledge extraction method which given small medical phrases extracts concept definitions, e.g., given "recent head injury" it would ideally extract the concept expression $\mathsf{RecentInjury} \sqcap \exists \mathsf{locatedIn}.\mathsf{Head}$.

Subsequently, we implemented a custom reasoner to compare such concept expressions with concepts in PGM. The need to implement a custom reasoner instead of an off the shelf one is motivated by the fact that although many of the ontologies we use are well-engineered, in many cases they contain "ill-defined" concepts whose meaning is still implicitly encoded in free text. For example, SNOMED does not define the concept $\mathsf{RecentInjury}$ in terms of Injury and Recent. Hence, the following subsumption cannot be identified by an OWL reasoner:

$$\mathsf{RecentInjury} \sqcap \exists \mathsf{locatedIn}.\mathsf{Head} \sqsubseteq \mathsf{Injury} \sqcap \exists \mathsf{occurred}.\mathsf{Recently} \tag{1}$$

Surprisingly, SNOMED contains a large number of such "ill-defined" concepts other examples of which are $\mathsf{ThermalInjury}$, which is not defined in terms of $\mathsf{Thermal}$ and Injury, $\mathsf{SevereDepression}$ which is not defined in terms of Severe and $\mathsf{Depression}$, $\mathsf{CardiacMuscleThickness}$, and more. Our hybrid reasoner is using the previous knowledge extraction method coupled with logic-based techniques to compare concepts w.r.t. subsumption. Another challenge for our reasoner was the scale of our KB which is at the order of half a billion triples and it is loaded to GraphDB. The issue is that like every other tripe-store, GraphDB is inherently incomplete for performing reasoning over constructors like those used in the above DL concepts, i.e., ObjectSomeValuesFrom. Hence, some extended DL inference functionality had to be simulated on top of GraphDB using SPARQL.

2 Extracting Concept Definitions From Text

To understand the meaning in text we built a method ($\mathsf{buildConcept}$) that extracts concept definitions from small medical phrases. Examples of such phrases are "acute duodenal ulcer", "granuloma surgical wound", "severe pain in left leg", etc. First, these phrases are decomposed into tokens using dependency parsing. Figure 1a depicts the dependency tree of the medical phrase "Recent pain provoked by injury". Nodes correspond to words in text and edges to linguistic relations between the tokens of the phrase. Second, a pre-processing step on the dependency tree is applied to obtain the graph depicted in Figure 1b. This graph contains mostly nouns, adjectives, and verbs while pronouns or other function words are suppressed. The nodes of the graph are then matched to classes from the KB by matching lemmatized text of each node to class labels or synonyms. Finally, the tree is traversed in a depth-first manner and class expressions are built. A non-trivial issue in our work is that the text rarely contains verbs which can be used as relations between classes as usually assumed in the literature [3, 4, 2]. For instance, in our running example the relation between "Recent" and "Pain" needs to be inferred. This is done based on the frequency that relations associate pairs of categories in the KB. For example, diseases are associated with temporal classes with the relation occurs. Hence, from the graph depicted in Figure 1b our method can construct the following class expression:

$$\mathsf{Pain} \sqcap \exists \mathsf{occurrs}.\mathsf{Recently} \sqcap \exists \mathsf{provokedBy}.\mathsf{Injury}$$

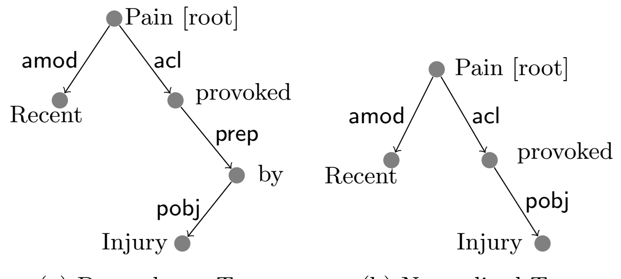


Fig. 1. Dependency and normalised trees of "Recent pain provoked by Injury".

3 Reasoning Using Textual Knowledge

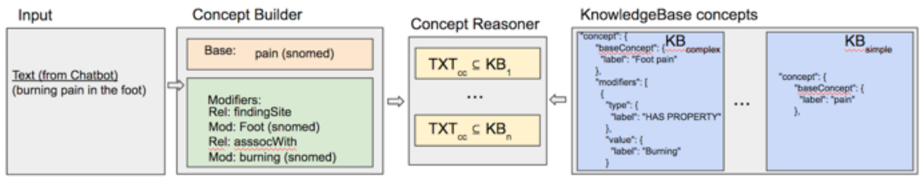
As mentioned in the introduction, the "undefinedness" of concepts in the KB motivated us to design a novel hybrid reasoning algorithm which given two concepts it exploits both semantic as well as textual information encoded in their labels to compare them w.r.t. subsumption. Given concepts C and D the algorithm briefly works as follows:

```
1. If isSubsumed$(C, D)$ then **return** true
2. $C^+ :=$ buildConcept$(C.label)$
3. If isSubsumed$(C^+, D)$ then **return** true
4. $D^+ :=$ buildConcept$(D.label)$
5. If isSubsumed$(C^+, D^+)$ then **return** true
6. **return** false
```

At step 1, the algorithm checks subsumption between two classes using a custom approximate algorithm which is based on a combination of structural subsumption [1], and SPARQL queries which attempt to simulate some of the consequence-based inference rules that are relevant to OWL constructors not handled by triple-stores like ObjectSomeValues. If subsumption fails, then it proceeds in trying to extract knowledge from the labels of the classes. For example, step 1. for the subsumption $\mathsf{RecentInjury} \sqsubseteq \mathsf{Injury}$ will fail and hence buildConcept("*RecentInjury*") would be called, returning $C^+ = \mathsf{Injury} \sqcap \exists \mathsf{occured.Recent}$. Then at step 3 $\mathsf{Injury} \sqcap \exists \mathsf{occured.Recent} \sqsubseteq \mathsf{Injury}$ returns **true**. The overall system (KAL) is depicted in Figure 2. User text is processed by conceptBuilder and the output is compared against PGM nodes using the reasoner.

4 Evaluation

To evaluate conceptBuilder alone, we randomly picked 200 classes from our KB which have a label that contains at least two words and used it to construct a concept definition. Out of the 200 concepts conceptBuilder failed to build a concept in 19 cases (9.5%) since it could not pick a concept from the KB. For



```
"concept": {
  "baseConcept": {
    "label": "Foot pain"
  },
  "modifiers": [
   {
    "type": {
     "label": "HAS PROPERTY"
    },
    "value": {
     "label": "Burning"
    }
```

```
"concept": {
   "baseConcept": {
    "label": "pain"
   },
```

Fig. 2. System overview

example, there is no concept for the word "Ionizing". For the remaining 181, we asked an in-house doctor to evaluate their quality using one of the labels wrong, correct, or partiallycorrect obtaining 21, 108 and 52, marked as such.

In the second experiment, we asked doctors to create natural language queries mimicking the text that users would type into the chatbot when they report their problem as well as the list of expected nodes form PGM that should be activated. For example, a user may type "*I cut my finger*", "*I feel really tired*", or "*my lower back hurts*" and the expected nodes are HurtFinger, Fatigue, and LowerBackPain, respectively; Table 1 presents Precision and Recall of our system and compared against a baseline approach that simply annotates medical terms using GATE annotator and then compares them to PGM as well as against a sentence embedding approach (emb) which has been trained using word embedding on medical blogs. As can be seen, using our system together with the embedder as a backup plan provides the best results.

Table 1. Precision and Recal of all chatbot reasoning approaches.

	GATE	GATE+emb	KAL	KAL+emb
Precision	0.57	0.74	0.77	0.84
Recall	0.51	0.73	0.79	0.88

References

1. Baader, F., Horrocks, I., Sattler, U.: Description Logics. In: van Harmelen, F., Lifschitz, V., Porter, B. (eds.) Handbook of Knowledge Representation, chap. 3, pp. 135–180. Elsevier (2008)
2. Buitelaar, P., Cimiano, P. (eds.): Ontology Learning and Population: Bridging the Gap between Text and Knowledge. Frontiers in Artificial Intelligence and Applications, IOS Press (2008)
3. Distel, F., Ma, Y.: A hybrid approach for learning SNOMED CT definitions from text. In: Proceedings of the 26th International Workshop on DLs (2013)
4. Gyawali, B., Shimorina, A., Gardent, C., Cruz-Lara, S., Mahfoudh, M.: Mapping natural language to description logic. In: Proc. of 14th ESWC. pp. 273–288 (2017)
5. Stoilos, G., Geleta, D., Shamdasani, J., Khodadadi, M.: A novel approach and practical algorithms for ontology integration. In: Proceedings of ISWC (2018)