Enhancing a Biomedical Ontology with Knowledge from Discharge Summaries

S.Després, C.Duclos, C.Le Duc, P.Vaillant

Université Sorbonne Paris Nord – LIMICS, Bobigny, France

{sylvie.despres,catherine.duclos,chan.leduc,vaillant}@univ-paris13.fr

Résumé

Pour effectuer des requêtes utiles et intéressantes sur les ontologies biomédicales en OWL avec un raisonneur, il est nécessaire de les peupler et d'ajouter de nouveaux axiomes. Ce travail vise à analyser les comptes rendus hospitaliers, disponibles sous forme de texte brut, en combinant une procédure déductive qui exploite une certaine régularité explicite dans leur structure avec un mécanisme d'apprentissage inductif. Les nouveaux axiomes ajoutés à la TBox et les assertions de la ABox résultant de ce processus d'analyse sont utilisés pour enrichir une ontologie biomédicale sélectionnée. Cela permet aux médecins, pharmaciens ou patients d'effectuer des requêtes sur l'ontologie enrichie en utilisant un raisonneur OWL afin d'obtenir des réponses, par exemple sur la façon dont les médicaments sont utilisés pour le traitement des maladies dans les hôpitaux.

Mots-clés

Ontologie, analyse de textes, peuplement, raisonnement

Abstract

To perform useful and interesting queries on biomedical ontologies in OWL with a reasoner, it is necessary to enhance them by populating directly their classes, properties, and adding new axioms. This work aims to analyse discharge summaries in raw text by combining a deductive procedure which exploits some explicit regularity in their structure with an inductive learning mechanism. New TBox axioms and ABox assertions resulting from this analysis process are used to enhance a selected biomedical ontology. This allows physicians, pharmacists or patients to perform queries on the enhanced ontology by using an OWL reasoner in order to obtain answers, for instance on how drugs are used for treatment of diseases in the hospitals.

Keywords

Ontology, analysis of texts, populating, reasoning

1 Introduction

Numerous biomedical ontologies have been designed to provide standardized terminologies used for recording clinical details of patients. SNOMED CT [9] is one of the most complete biomedical ontologies which covers various medical domains such as diagnostics, diseases, medications, anatomy, treatment. The main use of SNOMED CT so far is to offer to clinicians a nomenclature for creating medical documents which can be exchanged between different health care providers and researchers. However, the use of SNOMED CT as an OWL ontology allowing for powerful reasoning tasks remains very limited. For instance, one can straightforwardly use an OWL reasoner to check whether a class defined in the ontology is subsumed by another one, but it is less obvious to use such a reasoner to discover potentially conflicting portions of knowledge such as AbleToUseMedication, UnableToUseMedication, AllergicTo because of the absence of individuals and negated classes, properties from the ontology. This issue might be related to the modelling problematic [10] of individuals of classes such as Disease, Drug in an ontology. The tricky thing is that we cannot completely define a disease as an independent object, and there does not exist any nomenclature that assigns a name to every manufactured drug. As opposed to classes Disease and Drug, it is quite relevant to use names or the INSEE numbers (in France) to model individuals of class Patient. For instance, a clinician can use a convivial interface to add the following assertion in Description Logics [2], which is the basis for OWL, for stipulating that Anne is a patient:

$$\mathsf{Patient(Anne)} \tag{1}$$

where Patient is a class and Anne an individual. The point is that if Patient is populated then Disease and Drug are so *indirectly* thanks to the following OWL axioms:

$$\{\mathsf{Anne}\} \sqsubseteq \exists \mathsf{sufferFrom.\ Diabetes} \tag{2}$$

$$\{\mathsf{Anne}\} \sqsubseteq \exists \mathsf{medicatedWith.\ HumulinInsulin} \tag{3}$$

Axiom (2) tells us that Anne suffers from Diabetes. This means that there does exist an individual of class Diabetes which implies all symptoms specific to Anne's diabetes. Thus, this individual is indirectly created. In the same way, Axiom (3) defines indirectly without naming an individual of class HumulinInsulin which has all chemical and biological characteristics specific to the insulin taken by Anne.

It is common that the ontology is shared by several clinicians. Assume that one of them knows that Anne is allergic to HumulinInsulin. Therefore, she/he adds the following axiom for stipulating that Anne should not take any

$\mathsf{HumulinInsulin}$:

$$\{\mathsf{Anne}\} \sqsubseteq \forall \mathsf{medicatedWith}.\neg \mathsf{HumulinInsulin} \quad (4)$$

If two Axioms (3) and (4) are simultaneously present in the ontology, an OWL reasoner can discover inconsistency. This may prevent the clinician who tried to add Axiom (3) from introducing it to the ontology, and can help her/him to choose another insulin for Anne, for instance, IsophaneInsulin to which Anne is not allergic:

$$\{\mathsf{Anne}\} \sqsubseteq \exists \mathsf{medicatedWith}.\mathsf{IsophaneInsulin} \quad (5)$$

Note that the detected inconsistency is caused by both the conflicting knowledge as well as the presence of the individual Anne. Indeed, if the left-hand side of Axioms (3) and (4) is replaced by Patient, the ontology remains consistent.

We have shown an example where enhancing SNOMED CT by populating some classes can make it more exploitable in terms of reasoning. Such a task of enhancing SNOMED CT would require medical data sources which involve most basic concepts such as patient, disease, drug. A considerable volume of clinical knowledge is contained in electronic health records (EHR) in natural language. This knowledge is adapted to be read by a human, but is not easily adapted to tasks such as bulk querying, since the information is not structured. Extracting structured information from EHR is a task that has received a lot of interest from the medical knowledge engineering community over the past two decades. In this work, we use discharge summaries (DS), parts of EHRs, for enhancing SNOMED CT.

The main contribution of the present paper consists in proposing a method which extracts knowledge from medical texts, and formalizes the extracted knowledge as OWL assertions and axioms in order to enhance an existing biomedical OWL ontology. The feasibility of the method is established by experiments which consist of analyzing a dataset of 697 discharge summaries and generating about 10,000 OWL axioms and assertions. The obtained experimental results which are rigorously evaluated provide some hint on how to improve the method. We also present some use-cases to show how such an enhanced ontology can be exploited to answer DL queries with an OWL reasoner.

Named Entity Recognition and Linking in medical texts goes back to 1988 [6] and has been implemented by different systems using the basic bricks of Natural Language Processing (tokenization, sentence splitting, POS-tagging, stemming, gazetteer lookup): MetaMap [1], OBA [3], Bio-YODIE [7]. They were originally designed to perform very well on corpora of scientific literature, which generally feature standard text with few errors. Systems have also been designed with a focus on extraction information from EHRs, where sloppy syntax, abbreviations, and typos are frequent: cTAKES [13], MTCE [15]. Sytems using state-of-the-art NLP technologies, like cTAKES or most recent versions of MetaMap, perform fairly well on narrative text in normalized EHR, with an average F1-measure of 0.9 [12].

The NER feature implemented in spaCy [17, 16, 8] provides a powerful tool for analysing biomedical and clinical texts in order to recognize terms or concepts about diseases or medications. As any neuron-network based tool for NLP, spaCy relies on the notions of *word vectors* and *feature functions*. The former encodes different syntactical and semantic contexts where the words occur in the training texts while the latter tells NER how to predict assignments between entities and terms occurring in new texts by using word vectors. In this paper, we use spaCy to assign an abstract entity (label) to a *contiguous span of tokens* (terms) occurring in discharge summaries. For example, medicatedWith can be assigned to *penicillin, insulin, folic acid* if they occur in the medication section of a discharge summary.

2 The approach

The proposed method for enhancing a biomedical ontology with knowledge from discharge summaries consists of two stages corresponding to Deductive Component (DC) and Inductive Component (IC). The first stage extracts from each discharge summary the sections each of which refers to diseases, medications, or drugs to which the patient is allergic. This stage takes advantage of some explicit regularity in the structure of discharge summaries. The second stage uses spaCy to detect relationships from a patient to the diseases, allergies and medications recorded in the corresponding section extracted from her/his discharge summary in the first stage. In this second stage, we have chosen SNOMED CT to which Inductive Component adds OWL assertions and axioms created directly from the relationships previously detected such as patient-disease, patient-medications and patient-allergies.

The discharge summary of a patient is often written by the clinician who cared for the patient. It contains important information about the hospital visit of the patient such as (i) why she/he came into hospital, (ii) the results of any tests she/he had, (iii) the treatment she/he received, (iv) any changes to her/his medication. For example, most of discharge summaries in the dataset analysed in this paper contain several sections each of which refers to a kind of information such as identity of the patient, diagnostics on admission and discharge, drugs prescribed during hospitalization. Each section is usually started by a section title such as “PRINCIPAL DIAGNOSIS", “DIAGNOSTICS ON ADMISSION", “DISCHARGE MEDICATION". The body of each section is in raw text with eventual abbreviated terms used in medicine[1], for instance HTN stands for Hypertension. Note that the body of sections such as “DISCHARGE DIAGNOSTICS", “DISCHARGE MEDICATIONS" is freely written with irregularities and incoherent information. This makes analysis tasks for extracting concepts and terms much more challenging.

A Deductive Component takes discharge summaries in input and segments them into sections referring to diagnostics, medications and allergies. This segmentation is based on the fact that each such section starts with a specific section title (with some exceptions). Thanks to this reg-

[1] https://en.wikipedia.org/wiki/List_of_medical_abbreviations

```
RECORD #106886
ALLERGIES: Penicillin caused a rash.
PRINCIPAL DIAGNOSIS: STATUS POST
STAPHYLOCOCCUS ENDOCARDITIS
DISCHARGE DIAGNOSIS: INFECTION/RULE
OUT ENDOCARDITIS
...
DISCHARGE MEDICATIONS: Tylenol 650 mg
p.o.  q.4h.  p.r.n.  headache , ...
```

Figure 1: An extract from a discharge summary

ularity (that's why it is called deductive) in the structure of discharge summaries, we can develop scrips in Linux Shell/Python in order to extract expected sections from a discharge summary. For instance, if Deductive Component takes the discharge summary in Figure 1 as input, the following sections can be extracted :

```
RECORD #106886
```

```
ALLERGIES: Penicillin caused a rash.
```

```
PRINCIPAL DIAGNOSIS: STATUS POST
STAPHYLOCOCCUS ENDOCARDITIS
```

```
DISCHARGE DIAGNOSIS: INFECTION/RULE
OUT ENDOCARDITIS
...
```

```
DISCHARGE MEDICATIONS: Tylenol 650 mg
p.o.  q.4h.  p.r.n.  headache , ...
```

Extracted sections resulting from this component allow to reduce the search space for detecting relationships from a patient to the diagnostics and medications recorded in her/his discharge summary. Indeed, the contextual information would be independently found in each section, and thus, it is relevant that relationships of patient-medications should be extracted from the medication section rather than the diagnostics section.

An Inductive Component uses the statistical tool spaCy[2] (that's why it is called inductive) to detect relationships such as patient-diseases, patient-medications and patient-allergies from the sections extracted by Deductive Component. The most challenging feature of this task consists of irregularities in the way the interesting terms occur in the text. For instance, we can realize in the section DISCHARGE MEDICATIONS in Figure 1 that if we wish to detect drug names then only terms Tylenol, Ventolin, Beclovent are interesting. Moreover, when analyzing the section PRINCIPAL DIAGNOSIS in Figure 1, we expect a composite term such as STAPHYLOCOCCUS ENDOCARDITIS rather than one-word terms. In addition, if there is negated information included in a section such as RULE OUT ENDOCARDITIS in the section DISCHARGE DIAGNOSIS, then the term ENDOCARDITIS should be assigned to a negated entity.

To deal with these irregularities in the extracted sections, it is needed to use an approach which should not be based on deterministic or deductive principle but rather on an inductive method implemented in spaCy. Such a method relies on syntactical and semantic similarities between what it has learnt from training texts and what it meets in new texts. If a term occurring in a context learnt from training texts is assigned to an entity, then other terms occurring in the *same context* would be assigned to the same entity.

In order that spaCy predicts an assignment of a term included in an extracted section to an entity, it is needed to enhance a *core model* by using a set of training examples. Such examples are manually created by annotating *representative* sections selected from the extracted sections. The annotation of the text in an example tells spaCy which terms occurring in the text are assigned to an entity. When training a model for spaCy with several examples, it does not just memorize these examples but comes up to a theory which can be statistically generalized and used to predict similar assignments in new sections referring to the same topic. Figure 2 presents a training example in json format.

```
[ " DISCHARGE MEDICATIONS: Tylenol
650 mg p.o.  q.4h.  p.r.n.  ,
Amphojel 30 mL p.o.  t.i.d.  one-half
hour before meals , ciprofloxacin 500
mg ...",
{"entities":[[24, 31,
"medicatedWith"],[59, 67,
"medicatedWith"],[115, 128,
"medicatedWith"],[238, 248,
"medicatedWith"]]}]
```

Figure 2: A training example in JSON format for spaCy

In Figure 2, the raw text in the section DISCHARGE MEDICATIONS is manually annotated by indicating all drug names which should be assigned to the entity medicatedWith. For instance, the annotation `[24, 31, "medicatedWith"]` tells spaCy that Tylenol located between the indexes 24 to 31 should be assigned to medicatedWith. This example teaches to spaCy not only that Tylenol, Amphojel, ciprofloxacin,vancomycin should be assigned to medicatedWith, but also that other terms occurring in the text should not be assigned to medicatedWith. Moreover, if there are other terms (in other texts) occurring in the same context as that of Tylenol, Amphojel, ciprofloxacin,vancomycin, then they should be also assigned to medicatedWith. That means we need to choose *representative* examples among all extracted sections for training examples. Therefore, other training examples should be chosen such that drug names occur in a text with a different writing style and different vocabulary used. The annotation of representative training examples is a time-consuming task because spaCy would need a few hundred examples for training on a kind of text such as discharge summaries. Since the objective of the paper is just to show feasibility of our method, we have chosen and annotated about 50 representative sections for each entity sufferFrom, medicatedTo and allergicTo. These sections are extracted from about 696 discharge summaries.

The following figure shows the axioms and assertions ob-

[2] https://spacy.io

tained in Descition Logics which can be straightforwardly converted in OWL.

Patient(#161159)
acute_renal_ failure $\sqsubseteq$ Disease
$\{\#161159\} \sqsubseteq \exists$sufferFrom.acute_renal_failure
Patient(#432852)
vitamin_C $\sqsubseteq$ Drug
$\{\#432852\} \sqsubseteq \exists$medicatedWith.vitamin_C
Patient(#191371)
Penicillin $\sqsubseteq$ Drug
$\{\#191371\} \sqsubseteq \exists$allergicTo.Penicillin

To query on an OWL ontology, we can use an expressive query language [11] which allows variables to occur in queries. To the best of our knowledge, no engine which supports such a query language with an expressive fragment of OWL is available. To deal with DL queries on the enhanced ontology, we use a Web-based platform, namely ONTOREV [5, 4], based on HermiT [14] as OWL reasoner. This platform allows to perform some consuming-resource operations on large ontologies thanks to powerful servers. For instance, it offers to users an end-point for executing common DL queries that may require high performance in computation. The main goal of the present work is not about querying on a general OWL ontology. Therefore, we present briefly in this subsection some examples of queries which aim to raise awareness of the use of OWL ontologies in terms of reasoning. Since the full SNOMED CT is very large, we had to take from it just a small portion of the class hierarchy including only concepts related to the terms extracted from the discharge summaries. This makes experiments for reasoning on the enhanced ontology is possible. In the sequel, we present some interesting queries expressed in Description Logics which can be sent to the platform and get results via a Web interface. These queries can be straightforwardly translated to those in Manchester Syntax for testing with ONTOREV. First, if a clinician would like to know all patients who suffer from Diabetes and related diseases, then she/he can use the following queries.

Query 1:	Find all patients who suffer from diabetes
In DL query:	Find all instances of the concept $\exists$sufferFrom.Diabetes
Result:	There are at least 3 patients.[3]
Explanations:	If a patient #X is included in the result because there is an assertion $\exists$sufferFrom.Diabetes$(\#X)$ that was added to the ontology.

The clinician now wishes to check whether a patient was medicated with a drug to which she/he is allergic. For this, the clinician can find all patients who are allergic to for instance Aspirin, and display allergens related to Aspirin with the following queries.

Query 2:	Find all patients who are allergic to Aspirin
DL query :	Find all instances of the concept $\exists$allergicTo.Aspirin
Results:	There are at least 24 patients
Query 3:	Assume that #X is a patient noted in the result of **Query 2**. Find all drugs which are more general than Aspirin to which #X is allergic.
In DL query:	Find all super classes of Aspirin
Result:	Salicylate, Benzoic_acid Benzoic_acid_derivative Drug_allergen
Explanation:	The classes in the result are due to subsumption relationships defined in SNOMED CT.

Next, the clinician can use the following queries to display all patients who are allergic and medicated with Aspirin.

Query 4:	Find all patients who are allergic and medicated with Aspirin
DL query :	Find all instances of the concept $\exists$allergicTo.Aspirin$\sqcap$ $\exists$medicatedWith.Aspirin
Results:	There is at least one patient #24933
Query 5:	Assume that a patient who is allergic to a drug cannot be medicated with it. What happens in this case ?
Add axiom:	Is the ontology with the axiom $\exists$allergicTo.Aspirin $\sqsubseteq \forall$medicatedWith.$\neg$Aspirin is consistent ?
Result:	No
Explanation:	$\forall$medicatedWith.$\neg$Aspirin$(\#24933)$ contradicts $\exists$medicatedWith.Aspirin$(\#24933)$

All these queries can be reproduced with the Web-based platform accessible via `limicsb.univ-paris13.fr:8080/ontorev`. The ontology[4] is automatically loaded when clicking on "Load" button. With "More Infos" button, users can display the concept names, assertions, axioms, etc. of the ontology. With "DL query & Entailment" button, one can query on the ontology by using Manchester Syntax. For Query 5, users need to add the new axiom to the ontology with "Add New Infos" button, then click on "Add with Manchester Syntax" for entering the axiom, and check consistency of the modified ontology with "Check Consistency" button.

3 Experiments and Evaluations

In this section, we demonstrate feasibility of our method by experiments on a dataset of 696 discharge summaries

[3] There may be more patients in the results if the extracted terms are normalized. For example, we can observe that there are different terms such as Diabetes, DIABETES, Diabetes Mellitus, etc. which are extracted from the discharge summaries.

[4] The enhanced ontology is not downloadable since its content is built from the dataset made available by Harvard Medical School, which is not freely accessible,

from NLP Research Data Sets which are made available by Harvard Medical School[5]. From these 696 discharge summaries, Deductive Component extracted 646 diagnosis sections, 670 medication sections and 571 allergy sections. Since the allergic sections in the discharge summaries are usually short, almost allergens are correctly detected even if the negated information occurs in these sections such as “No known allergy", “The patient has no known drug allergies". We put in Figure 3 the number of axioms and assertions obtained by using our procedure over the sections extracted.

696 discharge summaries	Axioms	Asser.
DIAGNOSIS ON ADM/DISC	900	2,214
MEDICATIONS ON ADM/DISC	609	4,512
ALLERGY	372	878

Figure 3: The number of OWL axioms and assertions detected from the dataset.

We can observe that assertions are more many than axioms since a drug can be prescribed for several patients, or several patients can suffer from the same disease. In the sequel, we present separately the evaluations of diagnostics (diseases) and medications (drugs) extracted by the NER in spaCy (or spaCy for short).

For the evaluation of the results extracted from diagnostics sections, an expert takes randomly 50 from 696 discharge summaries in the dataset, read them blindly and extract from them all sections related to diagnostics. Then, the diagnostics extracted by the expert are compared with those extracted by spaCy. A *true positive* (TP) is a diagnostic extracted by spaCy and expected by the expert. A *false positive* (FP) is a diagnostic extracted by spaCy and not expected by the expert. A *false negative* (FN) is a diagnostic not extracted by spaCy but expected by the expert. The evaluation results of the diagnostic extractions are given in Figure 4.

50 discharge summaries	Expected	TP	FP	FN
Extracted Diagnostics	408	259	123	149
Extracted Medications	423	217	54	152

Figure 4: Results of extracted diagnostics and medications

Among 123 FPs detected by the expert, we can observe that

- there are 56 terms (diagnostics) which are wrongly decomposed such as “goiter" instead of “multinodular goiter", “type 2" instead of “diabetes mellitus type 2".
- there are 14 terms which are wrongly composed such as “hypertension morbid obesity" instead of 2 distinct diagnostics “hypertension" and “morbid obesity"
- there are 47 terms which are not diagnostics. For instance, ABVD is a treatment but not a diagnostic

For the evaluation of the results extracted from medication sections, the expert applies the same method as described above to 50 discharge summaries taken randomly from 696 discharge summaries in the dataset. The evaluation results of the medication extractions are also given in Figure 4.

	Precision	Recall	F-Mea.
Diagnostics Extraction	0.72	0.61	0.66
Medication Extraction	0.80	0.59	0.695

Figure 5: Recall and precision from the previous results

The low recall measure for the medications extractions might come from the fact that the extracted sections related to medications are not well delimited or they are truncated. A better performance may be expected if the borders of sections in a discharge summary are more precisely determined by the extractor in Deductive Component.

4 Conclusion and Future Work

We have presented a method for enhancing an OWL biomedical ontology by adding OWL axioms and assertions extracted from discharge summaries. Such a method makes the ontology more useful in terms of reasoning because users can query on it to get more information on relationships between medical concepts and to discover conflicting knowledge “encoded" in medical texts. This method can be applied to enhance OWL ontologies in another domain such as enhancing an ontology on (i) user profiles from Curricula Vitae or (ii) foods and nutrition from cooking recipes. The main idea is that a document like discharge summary is usually associated with an identifier that represents an object in real-world such as a patient, a candidate or a cooking recipe. Such a document should contain semantic relationships from this object to concepts described in the document. In this case, the structure of the document composed of different sections defines the semantics of these relationships. For instance, if a section such as “medications at discharge" is included in the discharge summary of a patient, then this implies a semantic relationship of “this patient is medicated with drugs given in this section".

Our method relies on Deductive and Inductive Components which use two complementary techniques to extract sections from each document, and detect concepts in each extracted section. The former takes advantage of explicit regularity in the structure of documents in order to segment them into portions in raw text while the latter uses a statistic approach to recognize terms in each portion whose meaning is related to the topic of the portion in question.

For future work, one of the most promising paths towards improving this method is modelling contextual information contained in the text. First, we will try to improve modelling temporal and modal context in order to get a better understanding of the patients' medical timeline, and be able to take into account mentions of risks or suspicion (instead of either erroneously counting them as positives, or discarding them and losing information).

[5] https://portal.dbmi.hms.harvard.edu/

References

[1] Alan R. Aronson and François-Michel Lang. An overview of MetaMap: historical perspective and recent advances. *J Am Med Inform Assoc*, 17:229–236, 2010.

[2] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, USA, 2nd edition, 2010.

[3] Mark A. Musen Clément Jonquet, Nigam H. Shah. The open biomedical annotator. In *Summit Transl Bioinform*, pages 56–60, 2009.

[4] Thinh Dong, Myriam Lamolle, Chan Le Duc, and Philippe Bonnot. Moteur de révision d'ontologie en SHIQ. *Ingénierie des Systèmes d Inf.*, 23(2):39–59, 2018.

[5] Thinh Dong, Chan Le Duc, and Myriam Lamolle. Tableau-based revision for expressive description logics with individuals. *J. Web Semant.*, 45:63–79, 2017.

[6] Peter L. Elkin, James J. Cimino, Henry J. Lowe, David B. Aronow, Tom H. Payne, Pierre S. Pincetl, and G. Octo Barnett. Mapping to MeSH. In *Proc Annu Symp Compu Appl Med Care*, pages 185–190, 1988.

[7] Angus Roberts Genevieve Gorrell, Xingyi Song. Bio-YODIE: A named entity linking system for biomedical text. `https://arxiv.org/abs/1811.04860`, 2018.

[8] E. Kiperwasser and Y. Goldberg. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327, 2016.

[9] Dennis Lee, Ronald Cornet, Francis Lau, and Nicolette de Keizer. A survey of snomed ct implementations. *Journal of Biomedical Informatics*, 46(1):87 – 96, 2013.

[10] Lennart Nordenfelt. On the ontology of disease. In *On the Nature of Health*, volume 26 of *Philosophy and Medicine*, pages 151–173. Springer, Dordrecht, 1995.

[11] Magdalena Ortiz and Mantas Simkus. Reasoning and query answering in description logics. In Thomas Eiter and Thomas Krennwallner, editors, *Reasoning Web. Semantic Technologies for Advanced Query Answering - 8th International Summer School Proceedings*, volume 7487, pages 1–53. Springer, 2012.

[12] Ruth Reátegui and Sylvie Ratté. Comparison of MetaMap and cTAKES for entity extraction in clinical notes. *BMC Medical Informatics and Decision Making*, 18(74), 2018.

[13] Guergana K. Savova, James J. Masanz, Philip V. Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C. Kipper-Schuler, and Christopher G. Chute. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *J Am Med Inform Assoc*, 17:507–513, 2010.

[14] Rob Shearer, Boris Motik, and Ian Horrocks. HermiT: A Highly-Efficient OWL Reasoner. In Alan Ruttenberg, Ulrile Sattler, and Cathy Dolbear, editors, *Proc. of the 5th Int. Workshop on OWL: Experiences and Directions (OWLED 2008 EU)*, Karlsruhe, Germany, October 26–27 2008.

[15] Charlotte Siefridt, Julien Grosjean, Tatiana Lefebvre, Laetitia Rollin, Stefan Darmoni, and Matthieu Schuers. Evaluation of automatic annotation by a multi-terminological concepts extractor within a corpus of data from family medicine consultations. *Int J Med Informatics*, 133(104009), 2020.

[16] Anders Søgaard and Y. Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In *ACL*, 2016.

[17] Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680. Association for Computational Linguistics, 2017.