

A System for Medical Information Extraction and Verification from Unstructured Text

Damir Juric,¹ Giorgos Stoilos,² Andre Melo,² Jonathan Moore,¹ Mohammad Khodadadi^{1*}

¹Babylon Health, London, SW3 3DD, UK

²Huawei Technologies, Research & Development, Edinburgh, UK

Abstract

A wealth of medical knowledge has been encoded in terminologies like SNOMED CT, NCI, FMA, and more. However, these resources are usually lacking information like relations between diseases, symptoms, and risk factors preventing their use in diagnostic or other decision making applications. In this paper we present a pipeline for extracting such information from unstructured text and enriching medical knowledge bases. Our approach uses Semantic Role Labelling and is unsupervised. We show how we dealt with several deficiencies of SRL-based extraction, like copula verbs, relations expressed through nouns, and assigning scores to extracted triples. The system have so far extracted about 120K relations and in-house doctors verified about 5k relationships. We compared the output of the system with a manually constructed network of diseases, symptoms and risk factors build by doctors in the course of a year. Our results show that our pipeline extracts good quality and precise relations and speeds up the knowledge acquisition process considerably.

1 Introduction

Significant effort and resources have been spent the last decade to encode medical knowledge using some semantically rich format like RDF and OWL leading to the creation of formal terminologies (Knowledge Based) like SNOMED CT, NCI, FMA, and many more. Despite these significant results, most of these KBs capture quite general medical knowledge while they contain few connections (relations) between their entities. For example, there is currently no comprehensive KB containing disease-symptom relations with only some preliminary and ad hoc manual efforts (Oberkampff et al. 2015; Mhadhbi and Akaichi 2017), not to mention other types of relations like risk factors, disease interactions, and more. Consequently, besides their use as vocabularies for text annotation and/or standardisation purposes such sources have not so far formed the basis for supporting diagnostic tasks for which purposes they need to be extended or specialised (Rubin et al. 2006).

To extract relations between medical entities (for either enriching existing KBs or creating new ones from scratch) many automatic approaches for medical Information (triple) Extraction (IE) have been proposed (Ernst, Siu, and Weikum 2015; Ruan et al. 2017; Shi et al. 2017). Most of them follow a supervised learning approach—that is, given a sentence like “Fever is a common symptom of Malaria” they first identify the medically relevant terms using a dictionary like UMLS and then train a classifier that associates the pattern that lies between these entities with the desired relation. Such approaches work well for small sentences which follow a kind of “definitional character”, however, in practice medical sentences can be much more complex like “The initial symptoms of Malaria are flu-like and include a high temperature”. In that case there is no pattern between the entities that implies the “symptom of” relation (the verb “include” is vague and can be associated with many different relations). In addition, most of these techniques require significant human intervention, fine-tuning, and lots of training data in order to learn good patterns that are able to differentiate between the relations. Moreover, new classifiers need to be trained whenever we want to extract triples involving a new relation.

In the current paper we present our system for extraction of relations between medical entities. This system acts as an enabler for other downstream systems in AI-enabled medicine that are supporting real-world use cases like symptom-checking, treatments, etc. Our approach is based on Semantic Role Labelling (SRL) (He et al. 2017) which has been used in the past with promising results (Christensen et al. 2011) and unlike pure ML approaches is fully explainable. Explainability of the system outputs is very important because users of the system are medical doctors. However, SRL is a general purpose method, hence several modifications and adaptations are required in order to improve its effectiveness. First, it is often the case that the relevant relation between two entities is “masked” by the presence of copula verbs (i.e., “is”, “are”)(Miwa et al. 2010). To address this issue we propose a method that is based on dependency parsing which adapts and extends the approach in (Miwa et al. 2010). Second, SRL-parsing is verb-centric while several relations in text are expressed through nouns (Yahya

*Work done while all authors were working for Babylon Health, London.

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

et al. 2014) (e.g., “symptom of”, “risk factor of”, etc.). To extract useful triples even in such cases a post-processing step is employed that replaces some verbs identified by SRL with nouns. Third, we investigate on techniques to filter, prune, and score the vast number of initially extracted relations in order to focus doctor verification only on those that are likely to be correct. In contrast to previous works we put significant focus not only on correct but also on medically informative triples. In many cases, correct triples like $\langle \text{Malaria spreadBy Mosquito} \rangle$ are extracted but these are not useful for clinical decision applications and in-house doctors would discard. For these purposes we developed a custom deep NN classifier which we evaluate obtaining encouraging results. Finally, for scoring triples we investigate on state-of-the-art Knowledge Graph embedding methods (Nickel et al. 2016) but also develop a custom ensemble one.

One of the roles of our system is to aid a computer supported medical diagnosis in Babylon Health¹. We have implemented our system, as a first step in this process, to extract information from unstructured data sources. System is currently in use by medical professionals to enrich an in-house medical KB and build and extend a symptom-checking engine. Provenance of each accepted fact is kept so it is possible to track back and explain decisions, an important requirement for systems certified as medical devices. Output of our system is: a) added to Babylon medical KB to be used for all other AI task requiring knowledge base (NLP, ML, inference, reasoning, etc) b) passed to a team of epidemiologists who score each triple by its epidemiological probability. In the last step, triples are consumed by the Bayesian network (Razzaki et al. 2018) (only correct triples are forwarded to this system) that in combination with a chat/dialogue system (for collecting specific information from a patient (Stoilos et al. 2019; Juric et al. 2018)) can finally provide possible diagnosis. All systems in this pipeline are equally important and in constant development in Babylon.

2 Knowledge Bases

Let \mathbf{C} and \mathbf{R} be disjoint sets of *concepts* and *properties*, respectively. Concepts and properties are uniquely identified using IRIs. A Knowledge Base (KB) is a tuple $\langle \mathcal{K}, \mathbb{T}, \mu \rangle$, where \mathcal{K} is a set of *subject, property, object* triples of the form $\langle s p o \rangle$ like in the RDF standard, \mathbb{T} is a subset of concepts from \mathcal{K} called *semantic types* (stys) and μ is a mapping from every concept in \mathcal{K} to a non-empty subset of \mathbb{T} . Semantic types denote classes that act as general/abstract categories of interest and are used to group other classes. In the following we will distinguish between *text* triples of the form $\langle \ell_1 \ell_2 \ell_3 \rangle$ where all ℓ_i are strings and are extracted in a first phase from unstructured text and *semantic* triples of the form $\langle C R D \rangle$ where C, R, D are entities (IRIs) from the KB and are triples that can be added in some KB. For simplicity we will simply write “triple” and whether it is a text or a semantic one would be clear by the use of notation.

¹<https://www.babylonhealth.com/>

3 Extracting Triples From Medical Sentences

Semantic Role Labelling (SRL) has been proposed in the past as a good approach for open information extraction (Christensen et al. 2011). This is because given a sentence it identifies its verbs, which usually correspond to relations, and then the arguments of the verb, which usually correspond to the subject and object of those relations.

Definition 1 Given a sentence *sent*, function `srlParse` returns a set \mathbf{F} of directed labelled graphs $\langle V, E \rangle$ (called frames) each one of which satisfies the following:

- it contains exactly one node $nd \in V$ (called main node) labelled $\mathcal{T}(nd) = \text{VB}$.
- every other node n is labelled with one of role types ARG0, ARG1, ARG2.
- every node is labelled with some string $\mathcal{L}(n) = \text{phr}$ that is a phrase from *sent*.
- for nd the main node, there is an edge $\langle nd, \text{arr} \rangle$ labelled $\mathcal{T}(\langle nd, \text{arr} \rangle) = \text{ARGUMENT}$ such that arr is an ordered list of nodes.

Only nodes labelled VB have edges labelled ARGUMENT. We call such a graph an SRL-graph of *sent*.

We focus on arguments of types ARG0, ARG1, and ARG2 since these roughly correspond to the thematic roles of PROTO-AGENT, PROTO-PATIENT, and INSTRUMENT, respectively, and hence capture the semantics of triples in the sense that some medical entity (disease, symptom, drug, organism) is causing some event or action to some other entity. We leave other types of nodes, like TEMPORAL for future work. A verb in a frame may have more than two arguments. Hence, since our goal is to extract triples (binary relations), the frames in an SRL-graph need to be scanned and take pairs of arguments with the main verb. This is accomplished using function `seek-get` defined next.

Definition 2 Given an SRL-graph \mathbf{F} , a role type A and a set of role types \mathbb{T} , function `seek-get`($\mathbf{F}, A, \mathbb{T}$) returns the smallest set of triples Tr such that if \mathbf{F} contains a node nd labelled VB with arr its arguments and $\text{arr}[i]$ and $\text{arr}[j]$ exist with $i < j$, $\mathcal{T}(\text{arg}_1) = A$ and $\mathcal{T}(\text{arg}_2) \in \mathbb{T}$, then Tr contains $\langle \mathcal{L}(\text{arr}[i]) \mathcal{L}(nd) \mathcal{L}(\text{arr}[j]) \rangle$.

In the next sections we highlight some practical issues and show how we propose to address them.

3.1 Resolution of copula verbs

It has been noted that the presence of copula verbs may degrade the quality of information extraction methods and removing them would be beneficial (Miwa et al. 2010). Consider, for example, the sentence “Optic neuritis **is** a common symptom that can **cause** blurred vision”. When applied on this sentence, SRL-parsing will compute frame F_1 with verb “is” and arguments “Optic neuritis” and “a common symptom that can cause blurred vision” and frame F_2 with verb “cause” and arguments “a common symptom that” and “blurred vision”. As can be seen, the relevant entities (“Optic neuritis”, “cause” and “blurred vision”) span across two different and disconnected frames. Hence, to be able to extract $\langle \text{OpticNeuritis causes BlurredVision} \rangle$ we need to propagate

Algorithm 1 eliminateCopulaVerbs(**sent**, **F**)

Input: A sentence **sent** and a set of frames **F** computed for **sent**.

- 1: Tree := dependencyParser(**sent**)
- 2: **if** $\langle n_1, n_2 \rangle \in \text{Tree}$ such that $\langle n_1, n_2 \rangle.l == \text{cop}$ **then**
- 3: Let F_{is} be the frame in **F** with the copula verb
- 4: Let F_{n_1} a frame different from F_{is} that contains $n_1.l$ in some argument Arg
- 5: If no such F_{n_1} exists then **return**
- 6: Let $\langle n_1, n' \rangle \in \text{Tree}$ such that $\langle n_1, n' \rangle.l = \text{nsubj}$
- 7: Replace Arg with the argument in F_{is} that contains $n'.l$ as a sub-string
- 8: Delete frame F_{is}

the main argument of the frame that contains the verb “is” to the respective argument of the second frame—that is, replace the argument “a common symptom that” in the second frame with “Optic neuritis”.

Differently than (Miwa et al. 2010) our approach, depicted in Algorithm 1, is more systematic and is based on dependency parsing. Given a set of frames **F** the algorithm tries to propagate information from one frame to the other when a copula verb is present. As a first step, it applies dependency parsing, a well-known type of syntactic parsing whose general properties we formalise next.

Definition 3 Let *phrase* be some text. Function *dependencyParser* takes *phrase* as input and returns a labelled tree, where each node (*nd*) is labelled with its corresponding word found in text (denoted by $nd.l$), each edge $\langle nd_1, nd_2 \rangle$ with one of the dependency relations (denoted by $\langle nd_1, nd_2 \rangle.l$), and all paths of edges $\langle nd_i, nd_{i+1} \rangle$ labelled with compound are collapsed to one node nd_1 with label $nd_1.l + “ ” + \dots + “ ” + nd_n.l$ where $+$ denotes string concatenation.

Note that we assume that the dependency parser groups all compound nouns. For example, we assume that if the function *dependencyParser* is applied on phrase “Heart Attack” it will return a single node with label “Heart Attack” rather than a tree with edge $\langle nd_1, nd_2 \rangle$ and labels $nd_1.l = \text{“Attack”}$, $nd_2.l = \text{“Heart”}$, and $\langle nd_1, nd_2 \rangle.l = \text{compound}$. This is done to assist the subsequent entity linking step and since such compound nouns are expected to appear in KBs.

3.2 Relations expressed through nouns

Although many relations are expressed using verbs, which favours the use of SRL, a large number is expressed using nouns. Examples are: *Common symptoms of malaria include fever, Anemia is a common symptom of sarcoidosis*, and more. Applied to the first sentence, the SRL-parser would return a single frame with verb *includes* and arguments *symptoms of malaria* and *fever* from which the triple *fever — includes — symptoms of malaria* can be extracted. In a similar way, from the second sentence the following triple would be extracted: *Anemia — is — a common symptom of sarcoidosis*. To be able to extract triples with “symptom of” as the property we employ simple yet effective patterns again based on dependency parsing which is inspired

Algorithm 2 nounRelationRewriting($\langle phr_1 \text{ verb } phr_2 \rangle$)

Input: A triple where *verb* is either “is” or “includes”.

- 1: Tree := dependencyParser(phr_2)
- 2: Let root be the root of the tree
- 3: **if** root is a noun **then**
- 4: Let *nd* be the closest descendant node of root that is also a noun
- 5: Replace *verb* with *root.l* and phr_2 with *nd.l*

Algorithm 3 tripleExtraction(**sent**)

Input: A sentence **sent**.

- 1: **F** := eliminateCopulaVerbs(**sent**, srlParse(**sent**))
- 2: candidates := seek-get(**F**, ARG0, {ARG1, ARG2}) \cup seek-get(**F**, ARG1, {ARG0, ARG2})
- 3: Triples := \emptyset
- 4: **for all** $\langle phr_1 \text{ verb } phr_2 \rangle \in \text{candidates}$ **do**
- 5: **if** *verb* $\in \{\text{“include”}, \text{“is”}\}$ **then**
- 6: nounRelationRewriting($\langle phr_1 \text{ verb } phr_2 \rangle$)
- 7: |\langle \text{link}(phr_1), \text{link}(verb), \text{link}(phr_2) \rangle
| |
- 8: Triples := Triples $\cup \{tr\}$
- 9: **end for**
- 10: **return** Triples

by the works in (Yahya et al. 2014). Our approach is depicted in Algorithm 2.

3.3 Information Extraction Pipeline

Based on the above our overall triple extraction pipeline is depicted in Algorithm 3. In brief, first SRL-parsing is applied, then copula verbs are eliminated and subsequently an initial set of candidate triples using the seek-get function is created. Next, the candidate triples are formed in the for-loop which also tries to apply the noun-rewriting algorithm. Finally, an IRI is associated from the KB using so-called entity linking techniques (Hachey et al. 2013) that try to associate a precise IRI from the KB to the phrases encountered in the text triples. At the present point we have experimented with a simple but effective approach that is based on a combination of text annotation with sentence embeddings (Zhelezniak et al. 2019; Cer et al. 2018). The latter approach maps sentences (or multi-world expressions) to a vector space with the property that (semantically) similar sentences are clustered closely in the space. Trained models can be used to compute embeddings for all labels of the KB as well as of the phrase that is encountered in the text and then their similarity can be estimated using angular distance. In order not to commit to a single IRI this step retrieves the top-*k* possible IRIs from the KB for every element which are later presented to doctors for selection.

4 Filtering, Scoring, and Validating

In the current section we present our approaches for filtering, scoring, and manually validating the extracted triples.

Our first filtering step is to focus only on a specific subset of properties as well as types that are of interest

in our use case. More precisely, only triples that contain one of the following properties: *symptom, cause, treat, affect, develop, lead, aggravate, alleviate, damage, trigger, spread, prevent, produce, misdiagnose, experience*, as well as triples where the subject and object is one of the following types: ClinicalFinding, Disease, Substance, Procedure, BodyPart, Animal, Behavior, ManufacturedObject, BiologicalFunction, BodySubstance, Organism, are considered.

Another issue is the extraction of correct but not precise triples like (Malaria spreadBy Mosquito) and (Swallow causing Pain). Since correct, the detection of such triples is a hard problem and a deep NN method was used that is described in the next sections.

4.1 Triple Scoring Model

The classifiers used by most IE frameworks associate some confidence score to the extracted triples which can be used for pruning or prioritising evaluation. To provide with similar scoring means we investigate the use of relational learning approaches for KBs (Nickel et al. 2016). Such approaches try to learn statistics about the structure of a given KB and then use the learned model to score a given triple. This score captures the degree to which each triple adheres to the model learned on by the KB structure.

We consider a set of different knowledge graph embedding models, such as TransE (Bordes et al. 2013), TransR (Lin et al. 2015), RESCAL (Nickel, Tresp, and Kriegel 2011), ComplEx (Trouillon et al. 2016), DistMult (Yang et al. 2015) and HoIE (Nickel, Rosasco, and Poggio 2016), as well as graph feature based approaches, such as SDValidate (Paulheim and Bizer 2014) and PaTy-BRED (Melo and Paulheim 2017), which use features that can be directly observed in the graph, e.g., types and paths. In an effort to combine the strengths of the two kinds of approaches, we also consider a relation-level stacking approach proposed by Meilicke et al. (2018).

All previous approaches were trained on a medical KB built in-house that integrates well-known terminologies like SNOMED CT, NCI, and more (Stoilos et al. 2018). A different meta classifier was learnt for each relation. To the best of our knowledge this is the first real-world use and evaluation of such techniques.

4.2 Candidate Triple Verification

Finally, all triples that have not been filtered out are candidates for manual verification by doctors. A triple may be incorrect because it implies medically incorrect information but also because the selected IRIs are wrong or don't capture the meaning of the text faithfully. Doctors verify all these aspects for each one of the subject (S), property (P), and object (O) of each triple separately. In case any of the tasks is rejected, then the whole triple is rejected. However, if doctors accepted any of the proposed entity mappings these acceptances will be saved and can be used in the future—that is, if some phrase *phr* has been linked to a concept *C* and the same phrase appears in another triple with the same attached concept *C* as proposed, the verification will be skipped with *C* as the linked correct concept to link to the phrase.

Relation	All	Acc.	Relation	All	Acc.
hasSymptom	45 669	1 059	cause	39 580	645
hasRiskFactor	2 152	115	treat	18 183	23
associatedWith	2 901	323	affects	5 031	-
hasComplication	7 060	256			

Table 1: Extraction stats; 120 606 total and 2 421 accepted.

5 Evaluation

Base on the above a triple extraction system was deployed in Babylon. For SRL parsing we used AllenNLP (He et al. 2017), for dependency parsing the StanfordNLP parser, for providing IRIs and structure for scoring we used a medical KB constructed in Babylon Health (Stoilos et al. 2018), (Barisevičius et al. 2018), while for IRI ranking we used Google's universal encoder (Cer et al. 2018).

5.1 Data Sources

Several textual data sources were analysed to decide which of them to use. Many previous works use PubMed, however, we decided that this data source is not focused on primary medicine but instead usually contains information about rare diseases. Moreover, in many cases, the results published there are at research level and verification and cross-validation by the wider community is still pending. Sources like MedlinePlus summaries and Wikipedia pages are more related to primary medicine. Sentence tokenization was applied on the sources and several meta-data like title of the document, source name, URL, time, and more were attached while bullet lists were flattened by converting them into a set of *n* sentences, for *n* is the number of bullets.

5.2 Triple Extraction Evaluation

We were running our pipeline through 2000 health topics/pages from MedlinePlus summaries and a Wikipedia dump. After collecting almost 900k triples and removing ones that did not contained any medical entities we ended with around 120K potentially interesting medical text triples. Table 1 presents the break-down of these triples according to a set of interesting properties. Doctors randomly start verification of knowledge in batches. Here we evaluate first batch that consists of around 5k relationships following the methodology described in Section 4.2. Out of these 5k triples, medical experts accepted 2421 which gives an acceptance rate of about 48%. Table 1 also presents a break-down of the accepted triples; no triple with relation “affects” has been accepted yet.

It is important to note that when a doctor rejects a triple this does not necessarily mean that the triple is wrong. In contrast, a triple may be factually correct but a doctor may think that it is not precise enough or relevant to the use case addressed by Babylon Health, like in the case of uninformative triples. Moreover, triples can also be rejected due to failures in entity linking. For the above reasons we feel that the above rate is in general not low. Nevertheless, after obtaining these initial sets of accepted and rejected triples we developed our imprecision predictor and scoring approaches that we briefly sketched previously. We used those methods

	SDValidate	TyBRED	PaTyBRED	TransE	DistMult	RESCAL	ComplEx	HolE	TransR	RelStack
AUC ROC	0.716	0.783	0.600	0.835	0.734	0.841	0.649	0.680	0.679	0.916

Table 2: Comparison of triple scoring approaches

to score triples and asked doctors to evaluate the top-500 scored ones. Out of the 500, doctors accepted 371 which gives an acceptance rate of 75% which shows that indeed these methods can prioritise triples that doctors would most likely accept. These scoring and triple predictor methods are described and evaluated individually in the following.

5.3 Triple Scoring Evaluation

In order to score triples we trained several latent and graph feature approaches on the Babylon Health KB. We also trained a relation-level stacking approach combining different triple scoring approaches. We then evaluated them by using as testing data accepted and rejected triples from the previous doctor verification step. From all approaches we tried we only report results for the top performing systems which where TransE (Bordes et al. 2013) and RESCAL (Nickel, Tresp, and Kriegel 2011), SDValidate (Paulheim and Bizer 2014), PaTyBRED and TyBRED (Melo and Paulheim 2017) from the class of graph-based approaches. For training the first two we used OpenKE² while for the latter two the implementation in (Melo and Paulheim 2017).

In the relation-level stacking approach (RelStack) we use logistic regression as the meta classifier. Different sets of triple scoring models have been tried and the best performing combination used set of widely distinct approaches comprising RESCAL, TransE, TyBRED and DistMult.

From doctor verification we obtained 1,642 accepted and 2,456 rejected triples; these are triples after the initial filtering of triples according to stys and relevant properties described in previous Sections. This set of triples is randomly split into a validation and testing sets, each containing half of the triples. The validation set is used to train the meta classifier in the stacking approach, and the testing set is used to compare the different methods. Table 2 shows the comparison results in terms of area under the ROC curve.

In an effort to detect uninformative triples we used the accepted and rejected triples produced in the verification step to train a classifier to predict which (subject, object) pairs bare possibly “interesting” information and which not. The subject and the object of each triple were embedded using the Universal Embedder (Cer et al. 2018) and the outputs were concatenated and presented to a deep learning model which consisted of 3 fully connected layers of 100, 100 and 30 neurons respectively, with ReLU activations. The final output is a single neuron with sigmoid activation. A dropout layer with 0.5 probability after the first layer and another with 0.2 probability after the second were also used.

For training, a set of 2,414 accepted and 2,414 rejected text triples was obtained. 80% of triples were used for training and the rest 20% for testing. We measured the accuracy by comparing the output of the classifier to the manually

Disease	triples	matched symp.	new symp.	new RF.	other
Malaria	132	38	24	3	25
Laryngitis	18	0	2	0	7
Urethritis	40	5	7	4	10
Atheroscl.	31	1	0	10	7

Table 3: Comparison of PGM with extracted triples.

assigned label by the doctors. We obtained an accuracy of 0.820911 and AUC ROC of 0.878573.

5.4 Evaluating Over a Symptom-Checking Network

Babylon Health develops a symptom-checking engine encoded as a probabilistic graph model (PGM) (Razzaki et al. 2018). This model consists of a highly interconnected graph with connections between diseases, symptoms, risk factors, and age groups and currently contains about 5k relations. These were created manually by doctors and epidemiologists based on medical textbooks, research papers, and statistical data in a course of a year. We wanted to check whether any portion of these relations were extracted by our pipeline. Such a check is not straightforward since the labels of entities in PGM may not be exactly the same as the strings in text triples. We used fuzzy string matching methods with a high threshold to do this comparison. From the 5k connections we found a close similarity for 2,5k of them. We asked a doctor to have a closer look into four diseases that exist in both PGM and our extracted set. The doctor was asked to check how many symptoms match and then whether new symptoms, risk factor, misdiagnosis, or relations to other diseases that are extracted by our IE could be interesting to them for extending and further developing PGM. The results are depicted in Table 3. From this analysis we concluded that the semi-automated process of fact extraction could help speed up the building of the PGM by suggesting facts that were overseen by domain experts and also help improve its quality and expand it with new relationships beyond its initial purpose like incorporating misdiagnosis and other types of relations.

6 Related Work and Conclusions

Several approaches for extracting clinical information from unstructured text have been presented in the past (Ernst, Siu, and Weikum 2015; Ruan et al. 2017). Most of them are based on supervised learning needing training data and adaptation when new properties are considered. Differently, in order to eliminate these needs we use Semantic Role Labelling as proposed in (Christensen et al. 2011). To improve the effectiveness of SRL in information extraction several novel techniques were proposed. First, a method to

²<http://openke.thunlp.org>

process and eliminate copula verbs, second, a lightweight method to process and explicate relations that are expressed through nouns and, third, methods for filtering and scoring triples with a particular focus on informative triples. The system was built and used by in-house medical doctors and the acceptance rate of the triples it generates is fairly high if one takes into account the high selectivity of doctors who can prune possibly correct but not useful triples. We also evaluated all our scoring models and showed that their use does increase the acceptance rate of triples. Finally, we also compared our extracted triples with an existing disease-symptom-risk factor network showing great overlap and potential usefulness of the former into enriching the latter.

References

- Barisevičius, G.; Coste, M.; Geleta, D.; Juric, D.; Khodadadi, M.; Stoilos, G.; and Zaihrayeu, I. 2018. Supporting digital healthcare services using semantic web technologies. In *ISWC*.
- Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *NIPS 26*, 2787–2795.
- Cer, D.; Yang, Y.; Kong, S.; Hua, N.; Limtiaco, N.; John, R. S.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; Sung, Y.; Strophe, B.; and Kurzweil, R. 2018. Universal sentence encoder. *CoRR* abs/1803.11175.
- Christensen, J.; Mausam; Soderland, S.; and Etzioni, O. 2011. An analysis of open information extraction based on semantic role labeling. In *K-CAP*, 113–120.
- Ernst, P.; Siu, A.; and Weikum, G. 2015. KnowLife: a versatile approach for constructing a large knowledge graph for biomedical sciences. *BMC Bioinformatics* 16:1–13.
- Hachey, B.; Radford, W.; Nothman, J.; Honnibal, M.; and Curran, J. R. 2013. Evaluating entity linking with wikipedia. *Artificial Intelligence* 194:130–150.
- He, L.; Lee, K.; Lewis, M.; and Zettlemoyer, L. 2017. Deep semantic role labeling: What works and what’s next. In *ACL 2017*, 473–483.
- Juric, D.; Stoilos, G.; Wartak, S.; and Khodadadi, M. 2018. Reasoning with textual queries: A case of medical text. In *ISWC*.
- Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, 2181–2187.
- Meilicke, C.; Fink, M.; Wang, Y.; Ruffinelli, D.; Gemulla, R.; and Stuckenschmidt, H. 2018. Fine-grained evaluation of rule- and embedding-based systems for knowledge graph completion. In *ISWC*, 3–20.
- Melo, A., and Paulheim, H. 2017. Detection of relation assertion errors in knowledge graphs. In *K-CAP*, 22:1–22:8.
- Mhadhbi, L., and Akaichi, J. 2017. DS-ontology: A disease-symptom ontology for general diagnosis enhancement. In *ICISDM*, ICISDM ’17, 99–102.
- Miwa, M.; Sætre, R.; Miyao, Y.; and Tsujii, J. 2010. Entity-focused sentence simplification for relation extraction. In *COLING*, COLING, 788–796.
- Nickel, M.; Murphy, K.; Tresp, V.; and Gabrilovich, E. 2016. A review of relational machine learning for knowledge graphs. *IEEE* 104(1):11–33.
- Nickel, M.; Rosasco, L.; and Poggio, T. 2016. Holographic embeddings of knowledge graphs. In *AAAI*, 1955–1961.
- Nickel, M.; Tresp, V.; and Kriegel, H.-P. 2011. A three-way model for collective learning on multi-relational data. In *ICML*, 809–816.
- Oberkampff, H.; Gojayev, T.; Zillner, S.; Zühlke, D.; Auer, S.; and Hammon, M. 2015. From symptoms to diseases — creating the missing link. In *ESWC*, 652–667.
- Paulheim, H., and Bizer, C. 2014. Improving the quality of linked data using statistical distributions. *IJSWIS* 63–86.
- Razzaki, S.; Baker, A.; Perov, Y.; Middleton, K.; Baxter, J.; Mullarkey, D.; Sangar, D.; Taliercio, M.; Butt, M.; Majeed, A.; DoRosario, A.; Mahoney, M.; and Johri, S. 2018. A comparative study of artificial intelligence and human doctors for the purpose of triage and diagnosis. *CoRR* abs/1806.10698.
- Ruan, T.; Wang, M.; Sun, J.; Wang, T.; Zeng, L.; Yin, Y.; and Gao, J. 2017. An automatic approach for constructing a knowledge base of symptoms in chinese. *Journal of Biomedical Semantics* 8(1):33.
- Rubin, D. L.; Dameron, O.; Bashir, Y.; Grossman, D.; Dev, P.; and Musen, M. A. 2006. Using ontologies linked with geometric models to reason about penetrating injuries. *AIME 2006* 37(3):167–176.
- Shi, L.; Li, S.; Yang, X.; Qi, J.; Pan, G.; and Zhou, B. 2017. Semantic health knowledge graph: Semantic integration of heterogeneous medical knowledge and services. *BioMed Research International* 2017(Article ID 2858423).
- Stoilos, G.; Geleta, D.; Shamdassani, J.; and Khodadadi, M. 2018. A novel approach and practical algorithms for ontology integration. In *ISWC*.
- Stoilos, G.; Wartak, S.; Juric, D.; Moore, J.; and Khodadadi, M. 2019. An ontology-based interactive system for understanding user queries. In *ESWC*.
- Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, E.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In *ICML’16*, ICML’16, 2071–2080. JMLR.org.
- Yahya, M.; Whang, S.; Gupta, R.; and Halevy, A. Y. 2014. Renoun: Fact extraction for nominal attributes. In *EMNLP*, 325–335.
- Yang, B.; Yih, W.; He, X.; Gao, J.; and Deng, L. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*.
- Zhelezniak, V.; Savkov, A.; Shen, A.; Moramarco, F.; Flann, J.; and Hammerla, N. Y. 2019. Don’t settle for average, go for the max: Fuzzy sets and max-pooled word vectors. In *ICLR*.