Knowledge Graph Construction For COVID-19 Domain

(Ngoc Khanh) Van Hoang
University of Stuttgart
Institution for Natural Language Processing
st171277@stud.uni-stuttgart.de

1 Introduction

1.1 Motivation

If knowledge discovery is about identifying potentially useful patterns in data, knowledge graph (KG) would be the representation of these patterns, structured into entities and semantic relationships between them. KG sits at the intersection of knowledge discovery, data mining, Semantic Web and Natural Language Processing (Kejriwal, 2019), allowing information to be reasoned and inferred about by computers. In other words, KG provides the kind of structured data, and factual knowledge, creating an impression of machines which possess intelligence and common sense similar to humans.

With the current pandemic COVID-19 effecting people' lives all over the world and creating a "new normal", one couldn't help but feel curious to learn more about it with any tools possible in hand. The motivation for this project is two-fold. First would be the desire to find out what scientists have discovered about this deadly virus in relation with other diseases and chemicals. The second motivation is to learn how to construct a domain-specific KG from scratch in order to facilitate a research need to collect information from unstructured text. Code for the project can be found here: `https://github.com/VanHoang85/Knowledge_Graph_Construction`.

The report is structured as follows: the next section gives a brief sketch of the steps and methods for constructing the KG and the related work we base our implementation on. Section 2 discusses in details individual tasks, as well as the setting for the experiment. Results are presented and discussed in Section 3. We also have another discussion on the use of CIDO as evaluation source in Section 4 before coming to our conclusion in Section 5. We also include a graph in Section 3, which is created from entities extracted from the COVID-19 dataset.

1.2 Related Work And Our Approach

In this project, we first aim to build a KG from a dataset of research papers on Covid-19. From that, we make a visualization of a subset of the graph which contains nodes related to the COVID-19 node and examine any the relationships hold among them. For the construction of the KG, we perform two main steps, entity extraction and relation extraction. We use *HunFlair* (Weber et al., 2020), a Named Entity Recognition (NER) system for biomedical texts to extract five different entities, namely cell lines, chemicals, disease, gene (or proteins), and species.

As for relation extraction, we consider it as the task of finding similar patterns between entities in a large text corpora via clustering technique. The underlying assumption is that pairs occurring in similar context belong to same clusters and thus share the same relations (Rozenfeld and Feldman, 2007). Our feature, or pattern, generation approach follows closely to that of Akbik et al. (2012) in which we exploit deep syntactic features. For any pair of entities appearing in a same sentence, we extract the shortest dependency path between them, together with other relevant non-path tokens to build patterns for our pair-pattern matrix.

Via clustering method, ideally, the clusters should carry three types of structured data which is highly desirable for our KG construction. First is the set of relations, which as the same as the set of clusters. Second is the set of entities belonging to each distinct relation. Finally, for each relation, we obtain the set of patterns describing the semantic relations held between entities in that cluster.

For evaluation purposes, we tried to perform both automatic and manual test on clustering quality. The Coronavirus Infectious Disease Ontology

(CIDO)[1] (He et al., 2020) is reserved for standard evaluation, using BCubed algorithm (Amigó et al., 2009). We could have taken advantage of this ontology earlier as an external resources in resolving relation issue between entities. However, it is often the case for a domain specific scenario that obtaining a high-quality, manually build ontology is a luxury. Therefore, our decision is to reserve CIDO for evaluation scheme in order to check whether methods developed for general domain dataset are suitable for specific biomedical domain. However, it unexpectedly turns out later that our automatic evaluation scenario has become an egg-and-hen problem. Therefore, we have performed a manual evaluation on the quality of the clusters. Additionally, a set of COVID-19 related entities are chosen randomly for visualization purposes. Graphs are to be created from these entities to show a potential relationship between COVID-19 and the extracted entities.

2 Knowledge Graph Construction and Evaluation

2.1 Dataset And Resources

The dataset used for the purpose of constructing the knowledge graph is a collection of research papers on the topic of COVID-19, assembled from various sources. From the filename, our assumption is that the dataset represents information and knowledge of the medical experts about the pandemic until March 20, 2020. Counting only those longer than five and shorter than forty tokens, the dataset consists of 5241859 sentences in total. In our experiment, we limit ourselves to the first 1000 sentences due to computational and memory constraints.

CIDO (He et al., 2020) is an ongoing community-driven open-source biomedical ontology in the area of coronavirus infectious disease. It is an effort to represent expert knowledge about the disease in a standardized, interpretable format for both humans and machine. The terms are collected from different standard ontologies, including Infectious Disease Ontology (IDO), Disease Ontology (DOID), Protein Ontology (PRO), Vaccine Ontology (VO), etc. As of September 25, CIDO contains 6503 classes and 446 instances, forming 115791 triples in total.

2.2 Entity Extraction

To extract entities, we use *HunFlair* (Weber et al., 2020), a pre-trained NER tagger for biomedical texts, developed under the Natural Language Processing (NLP) framework *flair* (Akbik et al., 2019). The tagger is trained on 23 biomedical NER corpora and can identify five different entity types, namely genes/proteins, chemicals, diseases, species and cell lines. To ensure that all COVID-19 entities are, without a doubt, able to be extracted, we add a simple rule that all matching tokens in the list {covid, covid-19, coronavirus, coronaviruses, corona, sars-cov-2 } will be tagged as COVID-19 entity.

For the extraction of entity pairs, we consider only those appearing in the same sentence. Pairs containing the same entities (for example, corona-corona) will be filtered out. This goes for nested pairs also as their dependency path will be empty and the optional tokens contains mostly irrelevant information about them. All entity pairs meeting these conditions will be treated as if they hold potential relationship for the next step, relation extraction. In other words, if a sentence consists of three entities, three possible pairs are taken into consideration: entity 1 and 2, entity 1 and 3, and entity 2 and 3.

2.3 Relation Extraction

Relation Extraction (RE) is the main focus of our project. We follow closely the approach of Akbik et al. (2012) to extract patterns held between entity pairs. Their work is based on a pair-pattern matrix to measure similarity between patterns via clustering methods to identify semantic relations (Rozenfeld and Feldman, 2007). Akbik et al. (2012) cluster entity pairs with cosine scores as distance metrics. For each entity pair, they collect core tokens on the shortest dependency path between these entities. Their algorithm then determines a set of optional tokens linked to core tokens with certain typed dependency. Features are generated from different combinations of these two sets. Similar to Wang et al. (2011), features are filtered out to eliminate patterns which are unlikely to represent a semantic relation.

2.3.1 Feature Generation Using Dependency Parsing

It has been shown that dependency paths are suitable as features for resolving ambiguity in relation extraction task (Mintz et al., 2009; Akbik

[1] https://github.com/cido-ontology/cido

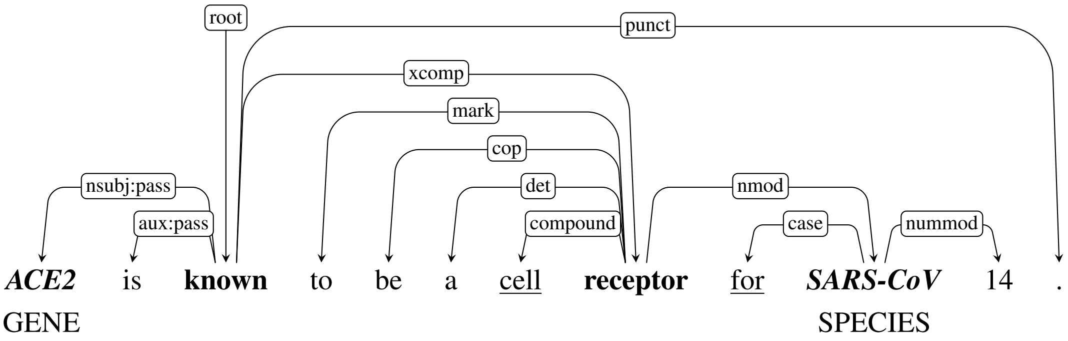

Figure 1: Dependency path of the example sentence. The entity pair and core tokens (shortest dependency path) are marked in bold while optional tokens are underlined.

feature type	generated features
1. SDP	{ known, receptor }
2. SDP + POS	{ know-VERB, receptor-NOUN }
3. SDP + entity types	{ Gene_1, known, receptor, Species_2 }
4. SDP + power set of optional tokens	{ known, receptor, cell }, { known, receptor, for }, { known, receptor, cell, for }

Table 1: Feature generation process for the sample sentence in Figure 1. One notice is that each feature is an unordered set of tokens which make up the pattern.

et al., 2012). Our implementation utilizes Stanza[2] (Qi et al., 2020), an NLP toolkit which incorporates syntactic analysis pipelines for biomedical texts. Its dependency parsers are trained on human-annotated treebanks, CRAFT and GENIA. Our experiment adopts the pre-trained parser on CRAFT treebank (Cohen et al., 2017).

For each pair of entities in a sentence, we compute their shortest dependency path (SDP). The tokens lying on this path are called *core tokens*, which are most likely to express a relation between two entities. To generate the set of *optional tokens*, all core tokens are examined. If their dependency relations to non-path tokens are meaningful, we add the non-path tokens to the set of optional tokens for this entity pair. The list of meaningful dependency relations is {compound, case, nsubj, acl, nmod}.[3]

The motivation for optional tokens is that some tokens, though not be a part of SDP, still contain important information. Akbik et al. (2012) gives an example of two entities *James Joyce* and *Nora Barnacle*, syntactically linked together by the token *and*. In other words, their SDP is *X and Y*, which is of no use to identify their semantic relation. However, by collecting the the token *married* linked to *James Joyce* by *nsubj* relation, the pattern *X and Y married* is meaningful enough for their clustering methods.

From the set of optional tokens, the power set is constructed and each combination of the SDP, except for the entity pair, and one set in the power set. Furthermore, following Akbik et al. (2013), we add another feature, including the SDP and entity types. Our experiment also includes indexes of two entity types to indicate which one comes first. The reason is because, different from other approaches, each of our features is actually a bag of tokens in which order is of no importance. Two entity pairs (COVID-19_1, Disease_2) and (Disease_1, COVID-19_2), though appear in similar context, might express two different semantic relations. Therefore, adding entity indexes is to make up for the word order which is lost due to our choice in constructing feature vector space.

Figure 1 shows an example sentence together with its dependency path, core tokens, and extra tokens. Table 1 illustrates how features are gen-

[2] https://stanfordnlp.github.io/stanza/available_biomed_models.html

[3] Our list is different from that of Akbik et al. (2012) as Stanza supports Universal Dependencies while theirs utilizes Stanford typed dependencies. Therefore, we converted their list to suit our experiment and also added two dependency relations which might be relevant to identify a relation.

erated from the sentence. Below is a summary of our feature generation criteria.

1. The shortest dependency path
2. The shortest dependency path and tokens' Part-Of-Speech
3. The shortest dependency path, and two entity types with their indexes
4. The shortest dependency path and one optional set drawn from the power set of optional tokens. Optional tokens are those in a certain dependency relation with core tokens.

Instead of using all features for clustering step, Wang et al. (2011) filtered out roughly 80% of the patterns which contain no semantic information and found an improvement in F1 scores. Similar to Akbik et al. (2012), we eliminate the features containing only closed word classes, or functioning words based on Universal POS tags[4] Our assumption is that a pattern without noun, verb, adjective, and adverb is simply of limited use to identify its semantic relation. Furthermore, we decide to limit features length to contain no more than 10 tokens after manually inspecting features for randomly picked sentences. Features longer than 10 tokens are mostly resulted from a long list of optional tokens. This list, if too long, usually leads to sparse matrix with hardly any match.

2.3.2 Clustering Entity Pairs With Similar Patterns

Our pair-pattern matrix has entity pairs as rows and features as columns with cell values being the number of co-occurrence counts. Cosine similarity is computed for distance metric (Bullinaria and Levy, 2007). In line with previous works in unsupervised RE (Rozenfeld and Feldman, 2007; Wang et al., 2011; Akbik et al., 2013), we perform Hierarchical Agglomerative Clustering clustering algorithm with the average linkage scheme (Han and Kamber, 2011) for merging clusters. Since no number of clusters is specified in advance, the algorithm requires a distance threshold above which clusters will not be merged. Through an exhaustive search, Akbik et al. (2013) suggested a high value (i.e. 0.999) for the threshold for good clustering results. *Scikit-learn* (Pedregosa et al., 2011) is our choice for the clustering implementation.

The obtained clusters represent distinct relations in the dataset. Additionally, we further collect entity pairs belonging to a certain relation and patterns potentially expressing that relation.

2.4 Evaluation Scenarios

Our initial plan is to use BCubed (Amigó et al., 2009), the most popular and effective algorithm to measure clustering quality. BCubed calculates the minimum intersection between two data points, taking into account only the fact that which items should belong together in a cluster. Cluster labels are disregarded. Thus, it is ideal for an unsupervised RE approach when the semantic relations are not clearly defined.

However, as it turns out later that we are unable to obtain ground-truth data from CIDO, we report only manual evaluation results in this report by examining all clusters having more than two pairs.

3 Experimental Results

number of ...	
entities	190
pairs	572
covid-related pairs	53
pairs with valid patterns	500
patterns	8098

Table 2: Statistics over entities, pairs, and patterns after two extraction steps

Table 2 shows the number of entities, pairs, and patterns after entity and relation extraction tasks. One note is that our experiment is performed on the first 1000 sentences only and not the entire corpus (5241859 sentences) due to limit on memory and computational power.

Among 572 pairs, the most frequently occurring one is *sars-cov* and *mers-cov* with 27 occurrences. The second and third pairs appear only 12 and 9 times respectively. And the number continues declining further and further. As for patterns, the extracted number is 8098, and the highest count is 30. However, the majority of patterns have less than three counts.

For clustering task, we perform a Agglomerative Clustering algorithm by scikit-learn library (Pedregosa et al., 2011) with parameters as stated in Section 2.3.2 with a high cutting threshold as suggested by Akbik et al. (2013). Following their

[4] https://universaldependencies.org/u/pos/all.html

entity pairs	patterns	count
('fever', 'dry cough'), ('fever', 'headache'), ('fever', 'pneumonia')	symptom	3
('hypertension', 'diabetes'), ('hypertension', 'cardiovascular disease')	smoking	3
('human', 'hcov-nl63'), ('human', 'hcov-hku1')	include	6
('covid', 'severe acute respiratory syndrome')	cause	4

Table 3: Some example entity pairs and their patterns

advice, we experiment with two values, 0.9 and 0.999, and obtain 291 and 250 clusters respectively. Counting the number of clusters having at least two pairs, we get 93 out of 291 clusters and 60 out of 250 clusters. The fact that the majority of clusters consists of one pair implies that our pair-pattern matrix is highly sparse.

Though our feature generation process imposes several restrictions, we believe sparse matrix is due to not enough data. In their experiment, Akbik et al. (2012) remove any pairs occurring fewer than 20 times. Among our entity pairs, only one meets this constraint. We could have lowered restriction on feature validity. For example, we could take into account all tokens connected to core tokens, not just potentially relevant ones in certain dependency typed. Or we could accept also features having no open world class tokens. However, Wang et al. (2011) see an improvement in F1 score despite filtering out 80% of irrelevant patterns, it raises doubt about lowering restriction on feature validity. Using more data and setting high restrictions might be a better alternative.

Manually checking clusters, we notice that although 0.9 value of cutting threshold could produce 93 candidate clusters, quite a few clusters contain pairs from the same sentence. Furthermore, a great number of clusters have pattern count of 1 and 2 only, making it virtually impossible to define any reliable relation between entities. On the other hand, 60 clusters obtained from 0.999 threshold produce fairly more refined patterns. Undoubtedly, the fewer the clusters are, the higher the pattern count should be, and thus the easier it is to determine which patterns are more relevant to expressing relation between entities. Table 3 illustrates example pairs and patterns.

3.1 Visualizing Extracted Entities

We use *graphviz*[5] library on Python to draw Figure 2. The graph is created from pairs having at least one entity belonging to COVID-19 type, regardless of their occurrence in the entire dataset. The graph shows relations between coronaviruses and several proteins/genes (DPP4, APN, Aminopeptidase N). Additionally, we can notice two connections about COVID-19. The first is about bat as the origin of the pandemic. And the other is pneumonia, an inflammatory condition of the lung: COVID-19 is well-known as an infection causing acute respiratory distress, which can lead to death.

4 Further Discussion: CIDO And Entity Identification

As briefly stated in Section 3, even though we could extract 7643 entities in total, we can obtain only 18 entities, using simple string match-up strategy. Our earlier hope is that medical terminology, for example gene and disease names, should be more straightforward and less ambiguous than general terms. This, unfortunately, turned out not to be true. Entity Linking (EL) task proves to be not a trivial issue. As of now, pre-trained EL models are for popular KG and ontologies only. As a result, doing an automatic evaluation becomes impossible as we found no matching pair between CIDO and ours.

In hindsight, the only way to avoid this issue is to take advantages of the terms from CIDO in an distant supervised manner instead of utilizing an NER system to search for entities. However, as mentioned in Section 1.1, one of our motivation for the project is to learn and understand the process of building a KG from scratch. Furthermore, we would like to reserve CIDO for purposes of evaluation. In retrospect, the issue of entity identification has become an egg-and-hen problem.

Regarding CIDO, the ontology is more like a collection of medical terminology related to COVID-19 than a network representation knowledge of the disease. Among 115791 triples, we obtain 7705 entities, making up 28353 entity pairs but only 40 valid relations. Here, we define "in-

[5] https://github.com/xflr6/graphviz

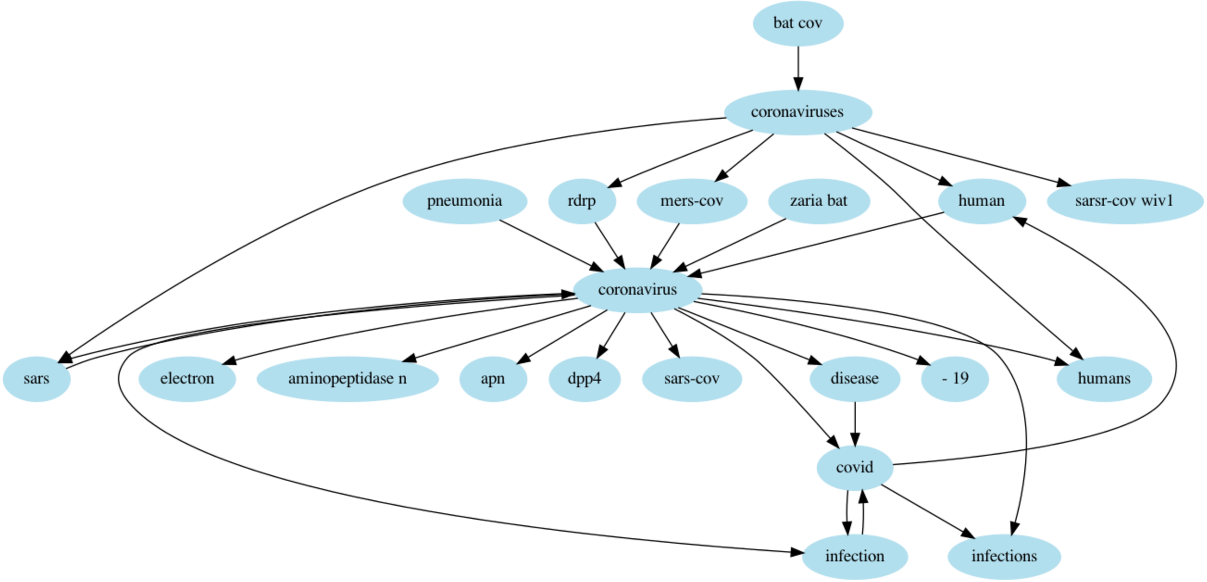

Figure 2: Visualize a graph from extracted entities

valid relations" as the ones which link an IRI of the term to its text or its undefined node. Upon closer inspection, among these 40 relations exist those such as *owl:inverseOf*, *uberon:present_in_taxon*, and *see also* and thus not suitable as semantic relations of medical entities.

When we attempted to look for triples with a valid relation *directly activates*[6], the search was unsuccessful. This term can only be found in either subject or object position, meaning that it is currently defined in super-relation or inverse relation to other semantic relations, not with entities. This leads us to the conclusion that CIDO is currently more like a term collection than a knowledge representation, and thus not suitable for KG evaluation purposes even if we can do a perfect Entity Linking task.

5 Conclusion

Our initial aim for the project is to understand the process of building a domain-specific knowledge graph from scratch. Also for this reason, we have decided to utilize entirely NLP tools for both entity and relation extraction tasks instead of relying on some handcrafted resources, from which we have learned a great deal. However, should we build another graph, or re-build the COVID-19 one, we would instead make use the information from CIDO as ground-truth entities. The reason is clear: NER taggers for biomedical texts are still far from perfect. Therefore, in a pipeline approach, errors from previous task can bias towards the following one. Needless to say, candidate patterns are to be less relevant. Furthermore, CIDO shows that more entities are known than relationships between entities. Also, knowledge graph is always about connection between entities.

As for relation extraction, the patterns are less refined than expected as the count in general is too low. Additionally, using clustering approach means the clusters are not easy to map into a well defined semantic relation. Nevertheless, it is too soon for any conclusion. From initial results, our hope is that should we be able to run the experiment on at least 10000 or 5000 sentences, we could be more stingy with setting requirements on minimum counts of pairs and patterns. A clear semantic relation could emerge accordingly. Also, in Section 4, we discussed at length our failed attempt to set up a standard automatic evaluation scenario. This raises the difficulty and challenge of evaluating a domain-specific KG as well as unsupervised relation extraction tasks.

In conclusion, the project has been a great learning experience. Except for NER and clustering, all other tasks and coding libraries and tools are completely new. Furthermore, it proves a chance to organize what has been discussed and presented during class, proving a big picture of how these tasks are linked together.

[6] http://purl.obolibrary.org/obo/RO_0002406

References

A. Akbik, T. Bergmann, Duncan Blythe, K. Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL-HLT*.

Alan Akbik, Larysa Visengeriyeva, Priska Herger, Holmer Hemsen, and Alexander Löser. 2012. Unsupervised discovery of relations and discriminative extraction patterns. In *COLING*.

Alan Akbik, Larysa Visengeriyeva, Johannes Kirschnick, and Alexander Löser. 2013. Effective selectional restrictions for unsupervised relation extraction. In *IJCNLP*.

Enrique Amigó, J. Gonzalo, J. Artiles, and M. F. Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12:461–486.

John A. Bullinaria and J. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39:510–526.

K. Cohen, Karin M. Verspoor, K. Fort, Christopher S. Funk, M. Bada, Martha Palmer, and L. Hunter. 2017. The colorado richly annotated full text (craft) corpus: Multi-model annotation in the biomedical domain. In *Handbook of Linguistic Annotation*, pages 1379–1394. Springer, Dordrecht.

Jiawei Han and M. Kamber. 2011. Data mining: Concepts and techniques, 3rd edition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Y. He, H. Yu, Edison Ong, Yang Wang, Yingtong Liu, Anthony Huffman, Hsin hui Huang, J. Beverley, J. Hur, Xiao lin Yang, Luonan Chen, Gilbert S Omenn, B. Athey, and B. Smith. 2020. Cido, a community-based ontology for coronavirus disease knowledge and data integration, sharing, and analysis. *Scientific Data*, 7.

Mayank Kejriwal. 2019. Domain-specific knowledge graph construction. In *SpringerBriefs in Computer Science*. Springer, Cham.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *ACL*.

Benjamin Rozenfeld and R. Feldman. 2007. Clustering for unsupervised relation identification. In *ACM*, page 411–418.

W. Wang, R. Besançon, Olivier Ferret, and B. Grau. 2011. Filtering and clustering relations for unsupervised information extraction in open domain. In *CIKM '11*, pages 1405–1414.

Leon Weber, Mario Sänger, Jannes Münchmeyer, Maryam Habibi, Ulf Leser, and Alan Akbik. 2020. Hunflair: An easy-to-use tool for state-of-the-art biomedical named entity recognition. *arXiv preprint arXiv:2008.07347*.