Relating Legal Entities via Open Information Extraction

Giovanni Siragusa[1], Rohan Nanda[1], Valeria De Paiva[2], and Luigi Di Caro[1]

[1] University of Turin - Department of Computer Science
Corso Svizzera 185, Turin, Italy
{siragusa, dicaro, nanda}@di.unito.it
[2] Nuance Communication Inc.
Sunnyvale, California, U.S.A
Valeria.dePaiva@nuance.com

Abstract. Concepts and relations within existing ontologies usually represent limited subjective and application-oriented views of a domain of interest. However, reusing resources and fine-grained conceptualizations is often challenging and requires significant manual efforts of adaptation to fit with unprecedented usages. In this paper, we present a system that makes use of recent Open Information Extraction technologies to unravel and explore corpus-centered unknown relations in the legal domain.

Keywords: Open Information Extraction · Natural Language Processing · Ontologies · Legal Concepts · Legal Text · IATE

1 Introduction

The Semantic Web research community needs tools for enriching and adapting existing semantic resources as well as for exploring relations within a given semantic resource and within a specific corpus. If we use such tools, extracted relations can be made then accessible to automatic systems or to domain experts in order to improve or support some particular work load. In this context, Open Information Extraction (OIE) systems [1] can be adopted to extract sets of triples of the form (*argument1*; *relational phrase*; *argument2*), where *argument1* and *argument2* are words (or multi-word expressions) and *relational phrase* is a phrase excerpt that describes the semantic relation between the two arguments.

In this paper, we present an OIE system, dubbed *LegOIE*, that automatically discovers concepts and relations in legal documents given a specific input ontology. The purpose of LegOIE is to enrich and adapt semantic resources, dynamically contextualizing concepts, browsing and providing other interactive facilities. Thus, we developed an OIE system that uses IATE, an European Union inter-institutional terminology database, to discover legal terms in the text and extract the phrase excerpt that connects two entities. Using a dictionary of legal terms improves the performance of the system since it will focus on specific entities, the legal ones. To prove this, we will compare LegOIE system with three state-of-the-art ones: Ollie [10], Reverb [6] and ClausIE [4].

2 Related Works

Open Information Extraction (OIE) was conceived to solve the problems of *Information Extraction*, which does not scale well in large corpora, where a huge set of relation is present. OIE have reached notable results in extracting relational phrases in large corpora such as Wikipedia and the Web [1, 12]. To the best of our knowledge, such systems are based on two steps: a tagging step where a Part-Of-Speech tagger or a dependency parser is applied to the sentence, and an extraction step that unravels the relational phrases. However, those systems can suffer from uninformative (relations which omit relevant information - for example, the triple (faust; made; a deal with the devil)) and incoherent (relations with no meaningful interpretation) extractions. Some research works have tried to solve this issue using heuristics. For instance, Reverb [6] uses syntactical constraints to filter relations, while Moro et al. [8] use a dependency parser and check if one of the arguments is marked as subject or object of a word in the relational phrase.

Differently from the previous systems, DefIE [5] constructs a syntactic-semantic graph by merging the output of the dependency parser with a Word Sense Disambiguation system. It extracts the relational phrases only between disambiguated words.

Other works used OIE systems to create or to populate ontologies and taxonomies. Nakashole et al. [9] applied OIE to automatically build a taxonomy, while Carlson et al. [3] and Speer and Havasi [11] used OIE to extend an existing ontology.

3 Resource, OIE system and Evaluation

In this section, we introduce the dictionary called IATE to label legal entities in running text and an Open Information Extraction (OIE) systems that use those tagged elements to extract triples. We will evaluate our system with three existing ones: Reverb, Ollie and ClausIE. Our evaluation seems to indicate that a system that uses a dictionary of legal terms can perform better than one that does not have such knowledge.

3.1 IATE Dictionary

The Inter-Active Terminology for Europe[3] (IATE), is the EU's inter-institutional terminology database, to discover concepts in the text. IATE consists of 1.3 million entries in English. Every entry (concept) in IATE is mapped to a subject domain. However, since some entities are wrong while others are entire sentences, we decided to filter some of these. First, we filtered out stopwords and concepts mapped to the "*NO DOMAIN*" label. Then, to find if a concept is related to a domain, we trained a word embedding using 2884 European Directives documents and 2884 Statutory Instruments documents. As word embedding model,

[3] See `http://iate.europa.eu/about\_IATE.html` for further details.

we used *fasttext*[4] [2] with an embedding size of 128 and default hyperparameters. This filtering phase was conducted using cosine similarity: we filtered all <term, domain> pairs that have a similarity lower then a given threshold. To choose the right threshold value, we manually constructed a developer set composed of 60 <term, domain> pairs manually extracted from IATE, equally divided in 30 pairs labelled as *incorrect* and 30 pairs labelled as *correct*. Then, we computed the cosine similarity between the embedding of the term[5] and the embedding of the domain, labelling as incorrect those entries that have a score lower than the threshold. Figure 1 shows the number of elements correctly recognized using different thresholds[6]. From the figure, we chose a 0.5 threshold (which is frequently used for this kind of tasks). After the cleaning, we obtained a dictionary composed of 37,158 entries.

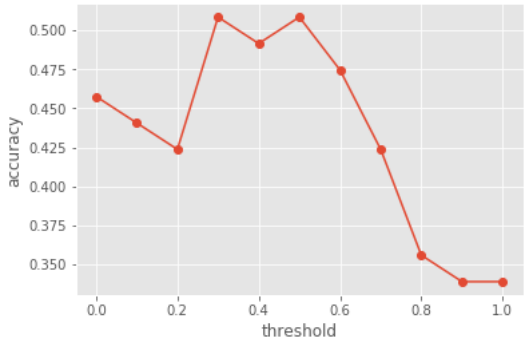

Fig. 1. The figure shows the accuracy for each threshold value. We experimented with a threshold in the range [0, 1], with a 0.2 step.

3.2 The OIE System

LegOIE takes as input a sentence and the IATE concepts that appear in the text, and returns a phrase excerpt for each possible pair of concepts. First, it processes the sentences through two steps: a Dependency Parser step and a Merging step. In the first step, Stanford CoreNLP [7] parser is applied on the sentences to generate a dependency graph. The graph is passed as input to the second phase, where words that form a single IATE concept are merged together. This steps allow to extract better relations that do not contain entity

[4] Fasttext, like Word2Vec, has both CBOW and SkipGram models. We used the CBOW since it performed well compared to the other one.

[5] In case that the term (or domain) is a multi-word expression, we represented it as the average of its word embedding.

[6] We recognized as correct an element that has a score lower than the threshold and that is labelled with *incorrect*.

words. Finally, LegOIE uses the list of entities in input to extract the shortest path that connects them in the undirected version of the graph. The output of the system is a set of triples of the form *(argument1, shortest path, argument2)* in which *argument1* and *argument2* are two IATE concepts. The extracted relations are then checked to see if they contain a verb (those not satisfying such condition are removed).

The extracted triples are then ordered according to their score. We computed the score of a triple using its frequency in the extraction and the length of the relation (number of words). Our intuition is to promote frequent triples with a short relation, while penalizing those that have a long relation name. Experience shows that long relations do not contain relevant verbal phrases that express a semantic relation (e.g., *made of* is a relevant relation). Our score formula is represented in Equation 1:

$$score(arg1, rel, arg2) = \frac{freq(arg1, rel, arg2)}{(H(rel) + 1)len(rel)} \tag{1}$$

where $freq(\cdot)$ calculates the frequency of the input, $H(\cdot)$ is the entropy of the relation, and $len(\cdot)$ calculates the length of the relation. We computed the entropy of the relation seeing how many times all the arguments that appear within that relation belong to the same IATE domain.

3.3 Evaluation

For the extraction, we tagged 4,310 documents containing European Directives (laws that all all European States have to implement) with the filtered IATE dictionary. We found that only 77,507 sentences contained at least two IATE concepts. Then, we applied LegOIE to extract triples, obtaining 2,267 such ones. We also extracted triples using Ollie, Reverb and ClausIE on the corpus. Those systems extracted[7] 3,060, 297,306, and 969 triples respectively.

Once we completed the extraction phase, we evaluated those systems on the base of their extracted triples: if the Open Information Extraction system could extract an informative triple where the two arguments are multi-word expression. In details, we randomly sampled 100 triples for each systems and we manually annotated them to check accuracy. Thus, we compute an accuracy score, calculating how many triples are labelled as correct. Table 1 shows the results of the evaluation, where we can see that LegOIE performed best, followed by Reverb. Furthermore, we can see that all OIE systems have a low accuracy, meaning that the task of extracting legal triples is challenging for automated systems.

4 Visualization

We decided to visualize the extracted triples in order to explore the relations extracted from the European Directives and how the concepts interact each

[7] ClausIE and Ollie stopped without completing the extraction due to an exception.

OIE system	Ollie	Reverb	ClausIE	LegOIE
Accuracy	0.17	0.21	0.13	0.32

Table 1. The table shows the four Open Information Extraction systems and their accuracy on 100 randomly sampled triples.

other. From the sorted triples, we selected the first 200 ones (those ones with a score equal or greater than 500). Then, we manually revised them, removing wrong ones. After this phase, we obtained 108 triples. We inserted those triples into GraphDB[8] to visually-explore them. GraphDB allows to search a relation, a domain or a an argument. Then, it is possible to expand a node to visualize its relations, navigating the graph and unveiling unknown semantic OIE-based interactions between concepts. This knowledge may be used to enrich the original resources. Figure 2 shows on the left the expanded graph created by clicking on the *animal* concept.

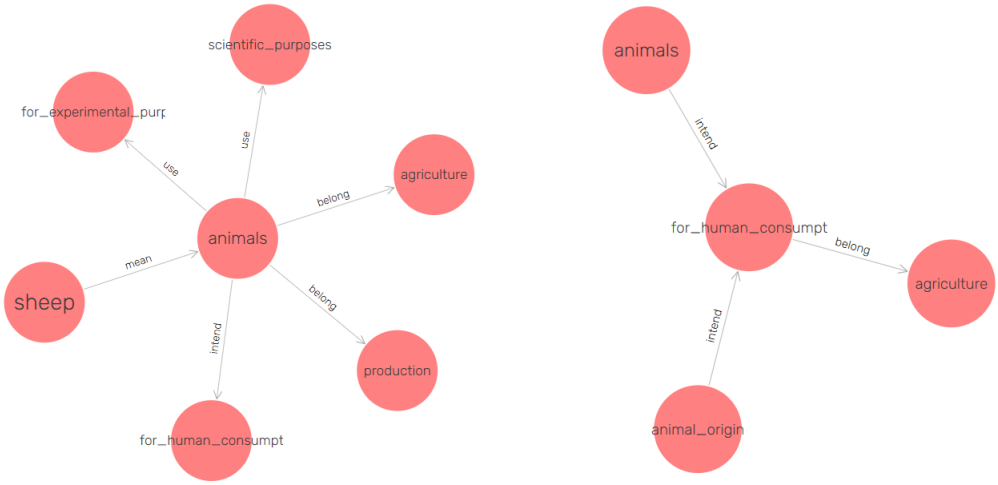

Fig. 2. The figure shows the entity *animal* and its relations with other entities. Each relation is represented by a labelled arrow. The label describes the relation type. The right graph reports the relations of the entity *for_human_consumption*.

5 Conclusion

We presented an Open Information Extraction-based system able to extract triples from a corpus containing concepts belonging to an existing input legal

[8] `https://ontotext.com/products/graphdb/`

ontology. The triples found by the system can be explored, discovering unprecedented interactions between the concepts. We compared our system with three existing OIE systems, founding that our performed well. As future work, we want to improve the Open Information Extraction system as well as the integrated visualization module, directing the investigation towards questions posed by subject experts interested in the contents of the European directives.

References

1. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the web. In: IJCAI. vol. 7, pp. 2670–2676 (2007)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
3. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr, E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: AAAI. vol. 5, p. 3 (2010)
4. Del Corro, L., Gemulla, R.: Clausie: clause-based open information extraction. In: Proceedings of the 22nd international conference on World Wide Web. pp. 355–366. ACM (2013)
5. Delli Bovi, C., Espinosa Anke, L., Navigli, R.: Knowledge base unification via sense embeddings and disambiguation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 726–736. Association for Computational Linguistics, Lisbon, Portugal (September 2015), `http://aclweb.org/anthology/D15-1084`
6. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 1535–1545. Association for Computational Linguistics (2011)
7. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The stanford corenlp natural language processing toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. pp. 55–60 (2014)
8. Moro, A., Navigli, R.: Integrating syntactic and semantic analysis into the open information extraction paradigm. In: IJCAI (2013)
9. Nakashole, N., Weikum, G., Suchanek, F.: Patty: a taxonomy of relational patterns with semantic types. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 1135–1145. Association for Computational Linguistics (2012)
10. Schmitz, M., Bart, R., Soderland, S., Etzioni, O., et al.: Open language learning for information extraction. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 523–534. Association for Computational Linguistics (2012)
11. Speer, R., Havasi, C.: Representing general relational knowledge in conceptnet 5. In: LREC. pp. 3679–3686 (2012)
12. Wu, F., Weld, D.S.: Open information extraction using wikipedia. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 118–127. Association for Computational Linguistics (2010)