The Workshops of the Thirtieth AAAI Conference on Artificial Intelligence
Knowledge Extraction from Text: Technical Report WS-16-10

Automatic Extraction of Events-Based Conditional Commonsense Knowledge

Arpit Sharma
Department of Computer Science
Arizona State University
Tempe, Arizona 85281
asharm73@asu.edu

Chitta Baral
Department of Computer Science
Arizona State University
Tempe, Arizona 85281
chitta@asu.edu

Abstract

Reasoning with commonsense knowledge plays an important role in various NLU tasks. Often the commonsense knowledge is needed to be extracted separately. In this paper we present our work of automatically extracting a certain type of commonsense knowledge. The knowledge resembles the kind that humans have about the events and the entities that participate in those events. One example of such knowledge is that *"**IF** A bullying B causes T rescued Z **THEN** (possibly) Z = B "*. We call this knowledge an event-based conditional commonsense. Our approach involves semantic parsing of natural language sentences by using the Knowledge Parser (K-Parser) and extracting the knowledge, if found. We extracted about 19000 instances of such knowledge from the Open American National Corpus.

1 Introduction & Motivation

In the recent years, NLU challenges have become more mainstream to AI. The challenges such as the Winograd Schema Challenge (WSC) (Levesque, Davis, and Morgenstern 2011) have been proposed to test the human level AI ability of systems. It is commonly acceptable that for understanding of text, one not only needs to give a semantic representation of the text but also needs to reason with various kinds of knowledge. In (Sharma et al. 2015) authors used one such knowledge in addressing a subclass of the WSC. This knowledge does not exist in the sources that are available today. In this work, we demonstrate our approach and experimental results of automatically extracting this kind of commonsense knowledge from a big natural language text repository. As a result of the extraction experiment about 2.8% of all the sentences in the corpus were found to contain the desired commonsense knowledge. The quality of the knowledge is also found to be good based on the qualitative evaluation (explained in section 5).

The popular NLU tasks such as hard co-reference resolution, and deep QA require some sort of world knowledge about the problem. Both factual knowledge such as *Gravity is a kind of force*, and commonsense knowledge such as *if there was no gravity then we would not stay on the ground* are important in completely understanding the meaning of text. A way in which such knowledge is attained by humans is by experiencing different scenarios that include entities (concrete or abstract), their interaction with other entities, and their participation in events (actions or state of being). There are systems such as IBM Watson for Deep Question Answering which make use of various Knowledge Bases (Singhal 2012; Leacock and Chodorow 1998) that contain such knowledge.

There has been various works for extraction of both factual and commonsense knowledge in the past such as WordNet (Miller 1995), ConceptNet (Liu and Singh 2004) and Cyc (cyc). Most of these are created by hand by a small number of people and hence it is not possible for them to have every kind of commonsense knowledge that every human has. Recently, there has also been some work for automatic extraction of knowledge from natural language text. One such attempt is NELL (Betteridge et al. 2009). It also does not focus on the kind of knowledge that we have extracted in this work. These knowledge repositories are a great source but as mentioned before, the knowledge contained in them is not sufficient to solve many hard NLU problems. For example the knowledge used in (Sharma et al. 2015) to address WSC is not present in any of them. This knowledge is of type, *"**IF** A bullying B causes T rescued Z **THEN** (possibly) Z = B"*.

In this article, we present our experiment to automatically extract this new type of commonsense from a natural language text repository. We discuss how this knowledge is different from the already existing ones, we explain the technical details of our approach and present detailed results of extraction experiments and provide an on-line interface to view this knowledge.

2 Our Approach

In this work, we automatically extracted a special kind of commonsense knowledge and created a database of public use. This knowledge is proved helpful in solving a subset of WSC (Sharma et al. 2015), which is a hard co-reference resolution challenge. Let us take an example inspired from WSC to better understand the knowledge. **Sentence:** *John was bullying Tom so we rescued him.* **Question:** *Who did we rescue ?* The Commonsense knowledge required to answer the question is:
IF A *bullying* B *causes* T *rescued* Z **THEN** (possibly) Z =

B. In other Words, **IF** *bullying* event *causes rescued* event **THEN (possibly)** *recipient* of *bullying* event = *recipient* of *rescued* event. This knowledge is based on the events (or actions) and their participants. Hence, we call is an Event-Based Conditional Commonsense (ECC).

In the sections below, we explain the main components of our knowledge extraction and retrieval system.

2.1 Corpus Selection & Semantic Parsing

Corpus We selected the freely available Open American National Corpus (OANC). OANC is a massive collection American English text, both written and spoken (for this work we chose only the written part of it). It is a collection of 15 million words, and it is designed to represent a wide cross-section of American English collected on or after 1990. The OAN corpus comes with different annotations but we have used the unannotated form of the written texts.

Semantic Parsing As we are interested in deeper semantics, we do not believe that they can be easily extracted from a bag-of-words or word-vector representations as it often ignores the context. To achieve the extraction of entities, events and their inter-relations, we used a semantic parsing and knowledge augmentation system called K-Parser (available at *www.kparser.org*) (Sharma et al. 2015). The K-Parser system translates a sentence into a semantic graph that captures the semantics in the sentence. This parser was developed as a part of the project (Sharma et al. 2015) to solve the Winograd Schema Challenge (WSC) (Levesque, Davis, and Morgenstern 2011) but it has evolved into a general purpose semantic parser since then.

The parser uses the Stanford Dependency Parser (De Marneffe and Manning 2008) as a base to retrieve the dependencies between the words in the input text. Due to the lack of generality in relations of Stanford Dependency parse, K-Parser uses a more general relation set from Knowledge Machine Component Library (Clark, Porter, and Works 2004), and many other new relations inspired from the need. The parser maps the Stanford dependencies to the semantic relations by using different rule based and classification algorithms. The output of K-Parser also has two level of conceptual class information about the events and entities in the input text. More information about K-Parser is available at *www.kparser.org*.

2.2 Knowledge Extraction

The goal of the Knowledge Extraction step is to identify and extract the relevant knowledge in a sentence. We took the logic-based approach for the extraction, where we see the knowledge extractor as an intelligent agent which uses a specific language to represent its knowledge and a reasoning algorithm to extract commonsense from text. We have used Answer Set Programming(ASP) (Gelfond and Lifschitz 1988; Baral 2003) to represent the agent's knowledge and to reason on the output of K-Parser. In the following sections we describe the representation of K-Parser output in ASP and the reasoning used by our knowledge extraction agent.

Representing K-Parser output in ASP We used RDF style representation for the output semantic graph (G) from the K-Parser. We took this step to make the graph comply with the syntax requirements of our logic programming module. Each edge in G is translated into a has-predicate. Each has-predicate is of arity three and has the following form $has(X, rel, Y)$, where X, Y are the nodes in the graph and rel is the edge label between X and Y. For example an edge labeled *"causes"* between *"bullying_3"* and *"rescued_7"* is represented as

```
has(bullying_3,causes,rescued_7).
```

Similarly all the edges are translated into has-predicates. The label *"bullying_3"* in the graph refers to the word *"bullying"* appearing at the third position in the input sentence.

Logical Reasoning The goal of the reasoning submodule is to find relevant knowledge from a set of has-predicates. To achieve this goal, we first encoded the domain knowledge in the agent's brain. The following block of code describes all possible relations between any two events in the K-Parser output.

```
eventRelations(causes;caused_by;
        enables;
        enabled_by;objective;
        next_event;previous;event;
        resulting_state;subevent;
        inibits;inhibited_by).
```

Knowing "all" possible relations between two event, an intelligent agent should be able to tell whether any given relation is an event relation. The ASP rule below encodes that information by using the *"nonEventRelation"* predicate for all the non event relations in the input graph.

```
nonEventRelation(R) :- has(X,R,Y),
              not eventRelations(R).
```

Similarly the following block of ASP code describes domain knowledge encoded in the agent's mind that two event nodes are connected via an event relation and a node is negative or positive.

```
relatedEvents(V1,R,V2) :- has(V1,R,V2),
                 eventRelations(R).
negative(V1) :- has(V1,negative,N).
positive(V1) :- not negative(V1),
                relatedEvents(V1,R,V2).
positive(V2) :- not negative(V2),
                relatedEvents(V1,R,V2).
```

Having this knowledge, the following block of code shows how the agent can extract relevant commonsense knowledge from the has-predicates.

```
answerEvents(positive,V1,VV1,R1,X1,R,
         positive,V2,VV2,R2,X2):-
         relatedEvents(V1,R,V2),
         has(V1,R1,X1),has(V2,R2,X2),
         has(X1,instance_of,X),has(X2,
         instance_of,X),has(V1,
         instance_of,VV1),
         has(V2,instance_of,VV2),
```

```
positive(V1),positive(V2),
nonEventRelation(R1),
nonEventRelation(R2).
```

When the predicate $answerEvents(.....)$ evaluates to $true$ for some assignment of the input variables according to the rule specified above, it describes that

"if event V1 is related to event V2 by an event relation R and the polarity of both the events are positive, then the entity X1 related to V1 with relation R1 is identical to the entity X2 related to V2 by the relation R2."

The values inside the predicate $answerEvents(.....)$ are (in order of occurrence) the polarity of the event V1, the actual value of event V1, base form of V1 i.e. VV1, relation between V1 and X1 i.e. R1, the actual value of X1, the relation between V1 and V2, the polarity of the event V2, the actual value of event V2, base form of V2 i.e. VV2, relation between V2 and X2 i.e. R2 and the actual value of X2.

The above block of code shows one ASP rule that is used to extract the commonsense in the case where both the related events are of positive polarity. The other three cases with different combinations of polarities also work in similar fashion.

2.3 Storage and Retrieval of the Knowledge

Storage This work was motivated by the application of commonsense knowledge in solving Natural Language Understanding problems such as hard co-reference resolution (Sharma et al. 2015). Hence, the goal here was to make the extracted knowledge available to the NLU research community so that it can be used in a variety of applications. To accomplish this we have used MongoDB database to save the extracted knowledge. MongoDB was chosen because of its speed, accessibility and usefulness.

Knowledge Retrieval The knowledge database consists of three sets of elements, namely, the set of events ($\mathcal{E}$), the set of relation among events ($\mathcal{R}$) and the set of slot-relations connecting events and entities ($\mathcal{S}$). Each query in the knowledge retrieval language posed to the database is a tuple consisting of elements from all these sets. Currently, we have defined the following seven queries,

1. *Event1*=E_1,*Rel*=R,*Event2*=E_2,*Slot1*=S_1,*Slot2*=S_2
2. *Event1*=*,*Rel*=R,*Event2*=E_2,*Slot1*=S_1,*Slot2*=S_2
3. *Event1*=E_1,*Rel*=R,*Event2*=*,*Slot1*=S_1,*Slot2*=S_2
4. *Event1*=E_1,*Rel*=R,*Event2*=E_2,*Slot1*=*,*Slot2*=S_2
5. *Event1*=E_1, *Rel*=R, *Event2*=E_2, *Slot1*=S_1, *Slot2*=*
6. *Event1*=E_1,*Rel*=$*$,*Event2*=E_2,*Slot1*=S_1,*Slot2*=S_2
7. *Event1*=E_1,*Rel*=$*$,*Event2*=E_2,*Slot1*=$*$,*Slot2*=$*$

The star ("*") in above queries means any legal value. For example, query 5 is used to extract all the knowledge instances where *"event1"* is E_1, *"event2"* is E_2, *"relation"* is R, *"slot1"* is S_1 and any legal value for *"slot2"*.

We have also developed a web interface for querying the knowledge base. The web application accepts the input query in both form-based and free-form (natural language). A demo version of the system is available at ***http://bioai8core.fulton.asu.edu/knet/***

3 Related Works

WordNet (Miller 1995) is one of the most popular knowledge base used by Natural Language Processing community. However, it is a lexical database consisting words, their senses, synonyms, hyponyms and hypernyms, it does not contain the commonsense knowledge that we are extracting in this work.

ConceptNet (Liu and Singh 2004) is another big source of commonsense knowledge. It is a semantic network containing more than 1.6 million edges connecting more than 300000 nodes where nodes represent concepts(words, small phrases) and edges represent the relation between nodes. However, the knowledge in ConceptNet is very high level and it does not have the kind that we are extracting in this work. Furthermore, the relations in ConceptNet are very coarse grained and also the participants of both the concepts are not specifically related.

Narrative Chains (Chambers and Jurafsky 2008), is another automatically extracted commonsense knowledge base. It contains a list partially ordered set of events that are centered around a common protagonist. The ordering of the events is temporal. Because of this, other relationships such as causality are not captured properly in narrative chains. An example of this is given in the *"bullying"* example mentioned in the sections above. In it there exists a causal relation between the events i.e. *"bullying"* causes *"rescue"*. In this example it seems obvious to say that the recipient of bullying is also the recipient of rescue but if we do not consider the causal relationship between events then the knowledge becomes less obvious.

Other popular knowledge bases such as Cyc (cyc) tend to compile complex assertions such as *every human has exactly one mother*. WebChild (Tandon et al. 2014) is a knowledge base created by extracting information from web. It contains properties of objects. For example *"Orange"* is *"round"* and its color is *"Orange"*. It lacks the constraint defined property knowledge such as the one mentioned in this work.

In addition to the above, there has been a lot of research towards building Knowledge bases with deeper knowledge instead of basic facts. One of the most interesting works are carried out by Dr. Oren Entzioni's group on Open Information Extraction. In (Lin, Etzioni, and Fogarty 2009), they focus on more interesting assertions such as *"the FDA banned Ephedra"* ignoring less useful statements such as *"the FDA banned products"*. In (Lin, Mausam, and Etzioni 2010), they have focused on commonsense knowledge inference based on the properties of relations such as functionality and transitivity. They have also proposed how to detect such properties, for example the functionality[1] of particular relations such as *bornIn*. Though different from our primary goals, their work on event extraction from twitter (Ritter et al. 2012) and entity linking in (Lin, Mausam, and Etzioni 2012) has inspiring thoughts in building higher-order knowledge bases in comparison to factoid ones.

[1]Functionality of a relation such as *bornIn* indicates that if (A,bornIn,B) and (A,bornIn,C) then B is same as C if they are either both locations or both time-range.

Table 1: Evaluation Results for 886 Randomly Selected Knowledge Instances

Rank	Evaluator 1 E_1	Evaluator 2 E_2 (in %)	Both E_1 and E_2
1	326 ($\approx$60%)	359 ($\approx$66%)	91 (45.5%)
2	54 ($\approx$10%)	44 ($\approx$8.1%)	15 (7.5%)
3	33 ($\approx$6.1%)	16 ($\approx$3%)	6 (3%)
4	19 ($\approx$3.5%)	30 ($\approx$5.5%)	3 (1.5%)
5	12 ($\approx$2.2%)	21 ($\approx$3.9%)	3 (1.5%)
6	99 ($\approx$18.2%)	73 ($\approx$13.5%)	38 (19%)
Miss-Matches	-	-	44 (22%)
Total	543	543	200

4 Evaluation

The kind of knowledge extracted here is already proved useful in solving a hard problem (Sharma et al. 2015) such as WSC. Now the questions arise, if the knowledge base created in this experiment has sufficient instances of such knowledge and are those instances are of any use. Both quantitative and qualitative analysis were performed to address these questions. The details are mentioned below.

We performed the knowledge extraction experiment on the Open American National (OAN) corpus. The written part of the corpus contains a total of 6405 documents from six genre. On an average each document contains 106 sentences. Out of all the sentences our system was able to extract 19336 instances of knowledge. In other words, about 2.85% (19336 out of 678930) of the sentences in OAN corpus were found to contain the desired commonsense knowledge.

For the qualitative analysis, a set of 886 instances of the commonsense knowledge were randomly sampled from the 19336 total instances. Two human evaluators[2] were employed to test the quality of the instances. Each evaluator was provided with a natural language translation of 543 commonsense knowledge instances (with an overlap of 200) along with the instructions to rank the quality of the knowledge. This evaluation instructions were inspired from (Gordon, Van Durme, and Schubert 2010). The table 1 shows the evaluation results. Here, rank 1 means the evaluator agrees that the knowledge instance is clear and entirely plausible. Rank 6 means the evaluator disagrees on the good quality of the knowledge. Rank 2 to 5 are in the order of decreasing evaluator agreement.

We also analyzed the agreement among the evaluators by counting the difference in the rankings of both the evaluators. The results' chart is shown in Fig. 1. From this experiment, we found that the evaluators were in agreement 156 of the times out of 200 and 25 times there was only a difference of 1 between their rankings (cased on our evaluation schema). This shows that most (90.5%) of the times the evaluators were either in complete agreement with each other or they ranked next to each other.

[2] One undergraduate Computer Science student and one graduate Computer Science student.

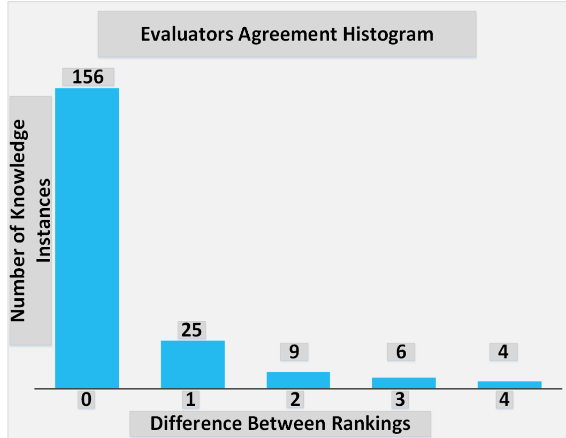

Figure 1: Evaluators Agreement Chart

5 Conclusion & Future Work

In this paper, we presented our work on using a semantic parsing and logical reasoning technique to extract a new kind of commonsense knowledge from text repository. The knowledge extracted has already been proved useful in solving an NLU task. Also, this knowledge is not present in currently available knowledge bases. The quantity of knowledge extracted is noticeably less than the some other similar works such as KNEXT (Van Durme and Schubert 2008), where there are about 1.78 unique knowledge instances extracted from each sentence in the corpus. This is because the kind of knowledge that we are extracting in this work is of very different nature and significance. Also, the knowledge we are extracting is based on the experience earned by people over the years and it is very difficult for a small group of people to list such knowledge by hand. Furthermore, the quality of the knowledge is determined in a fashion similar to KNEXT and found to be much better ($\approx$ 67% as compared to 54% in KNEXT) .

Though, this project was aiming at extraction of a specific commonsense knowledge, the ultimate goal is to construct a knowledge-base which will concentrate more on "knowledge" than facts. This "knowledge" points to higher-order relations (or rules in terms of Formal Semantics) than the ones which can be expressed using ontological predicates such as *isA* and *hasA*. As demonstrated in this work, such knowledge is better captured by the relations between entities and events. As future work, there is a need to define more types of commonsense knowledge based on relations among multiple events and the procedure to extract such knowledge automatically from textual corpora. Also, there is a scope of research on whether other types of relations are required to represent the complete domain of commonsense knowledge; and if yes, such types must be identified.

Acknowledgements

We thank NSF for the DataNet Federation Consortium grant OCI-0940841 and ONR for their grant N00014-13-1-0334 for partially supporting this research.

References

Baral, C. 2003. *Knowledge representation, reasoning and declarative problem solving*. Cambridge university press.

Betteridge, J.; Carlson, A.; Hong, S. A.; Hruschka Jr, E. R.; Law, E. L.; Mitchell, T. M.; and Wang, S. H. 2009. Toward never ending language learning. In *AAAI Spring Symposium: Learning by Reading and Learning to Read*, 1–2.

Chambers, N., and Jurafsky, D. 2008. Unsupervised learning of narrative event chains. In *ACL*, 789–797. Citeseer.

Clark, P.; Porter, B.; and Works, B. P. 2004. Km - the knowledge machine 2.0: Users manual. *Department of Computer Science, University of Texas at Austin*.

Cyc Knowledge base. Available at *http://www.opencyc.org/doc*.

De Marneffe, M.-C., and Manning, C. D. 2008. Stanford typed dependencies manual. *URL http://nlp. stanford. edu/software/dependencies manual. pdf*.

Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. In *ICLP/SLP*, volume 88, 1070–1080.

Gordon, J.; Van Durme, B.; and Schubert, L. K. 2010. Evaluation of commonsense knowledge with mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, 159–162. Association for Computational Linguistics.

Leacock, C., and Chodorow, M. 1998. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database* 49(2):265–283.

Levesque, H. J.; Davis, E.; and Morgenstern, L. 2011. The winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*.

Lin, T.; Etzioni, O.; and Fogarty, J. 2009. Identifying interesting assertions from the web. In Cheung, D. W.-L.; Song, I.-Y.; Chu, W. W.; Hu, X.; and Lin, J. J., eds., *CIKM*, 1787–1790. ACM.

Lin, T.; Mausam; and Etzioni, O. 2010. Commonsense from the web: Relation properties. In *AAAI Fall Symposium: Commonsense Knowledge*, volume FS-10-02 of *AAAI Technical Report*. AAAI.

Lin, T.; Mausam; and Etzioni, O. 2012. Entity linking at web scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, AKBC-WEKEX '12, 84–88. Stroudsburg, PA, USA: Association for Computational Linguistics.

Liu, H., and Singh, P. 2004. Conceptneta practical commonsense reasoning tool-kit. *BT technology journal* 22(4):211–226.

Miller, G. A. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Ritter, A.; Mausam; Etzioni, O.; and Clark, S. 2012. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, 1104–1112. New York, NY, USA: ACM.

Sharma, A.; Ha, N. V.; Aditya, S.; and Baral, C. 2015. Towards addressing the winograd schema challenge - building and using a semantic parser and a knowledge hunting module. In *Proceedings of Twenty-Fourth International Joint Conference on Artificial Intelligence*. AAAI.

Singhal, A. 2012. Introducing the knowledge graph: things, not strings. *Official Google Blog, May*.

Tandon, N.; de Melo, G.; Suchanek, F.; and Weikum, G. 2014. Webchild: harvesting and organizing commonsense knowledge from the web. In *Proceedings of the 7th ACM international conference on Web search and data mining*, 523–532. ACM.

Van Durme, B., and Schubert, L. 2008. Open knowledge extraction through compositional language processing. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, 239–254. Association for Computational Linguistics.