Learning from Reading Syntactically Complex Biology Texts

Rutu Mulkar, Jerry R. Hobbs and Eduard Hovy
Information Sciences Institute, University of Southern California
$\{rutu, hobbs, hovy\}$@isi.edu

Abstract

This paper concerns learning information by reading natural language texts. The major aim is to develop representations that are understandable by a reasoning engine and can be used to answer questions. We use abduction to map natural language sentences into concise and specific underlying theories. Techniques for automatically generating usable data representations are discussed. New techniques are proposed to obtain semantically correct and precise logical representations from natural language, in particular in cases where its syntactic complexity results in fragmented logical forms.

Introduction

It has long been a vision of artificial intelligence to build large knowledge bases by automatically reading instructional, natural language texts and converting them into axioms in a logical theory. This requires a base of commonsense knowledge — the knowledge a human must bring to the instructional text — and the natural language processing techniques for interpreting text and translating it to a usable logical representation, and the ability to integrate the logical form into the knowledge base. This paper describes a project that has focused on the language interpretation problem. It is our second experiment in learning by reading scientific texts. The first focused on chemistry textbook descriptions of chemical reaction (Mulkar *et al.* 2007). In this work we are using the biology domain for our experiments, focusing on a set of paragraph-length texts describing the heart, obtained from a variety of sources, including Wikipedia and other webpages.

The Halo Project (Friedland & Allen 2004) evaluated various existing knowledge representation and reasoning techniques and demonstrated their utility in the learning by reading task, once the text is translated to logic. In our work we develop a front end to these KRR models and automate the generation of the required semantic structures from natural language sentences. This provides an automated flow, entirely independent of human intervention, from natural language text to a deep representation that enables question answering. The quality of learning performed by the system has been evaluated by asking questions regarding the text and evaluating the amount of information that has been learned from it.

In this work, we link to the KM (Clark, Harrison, & Thompson 2003) system, which records the derived logical form triples (represented as a growing, interconnected network of expressions), integrates them into the growing model of what has been read so far (which might require inference to obtain some interconnections), performs additional inference using existing axiomatic knowledge (which adds further expressions and interconnections), and determines answers to questions.

In this paper we describe the successive stages in handling the natural language, then focus in particular on our use of abductive inference to recognize semantic linkages that syntactic analysis fails to provide.

Language Processing

The text is parsed, a shallow logical form is generated, and abductive inference is used to convert the content of the text to the language required by the KM (Knowledge Machine) (Clark, Harrison, & Thompson 2003) system for matching its models of devices. We here refer to this entire process as NL.

Parsing

After initial text cleanup and occasional creation of lexical items, each sentence of the text is parsed using the Contex parser (Hermjakob & Mooney 1997). This parse is used to create a shallow logical form as explained below. A sample Context parser output is:

```
[1] The heart is a pump [S-SNT]
     (SUBJ) [2] The heart [S-NP]
                  (DET) [3] The [S-DEF-ART]
                  (PRED) [4] heart [S-COUNT-NOUN]
     (PRED) [5] is [S-AUX]
     (OBJ) [6] a pump [S-NP]
                  (DET) [7] a [S-INDEF-ART]
                  (PRED) [8] pump [S-NOUN]
```

Table 1: Sample output of Contex Parser

The actual output obtained after the parse is a considerably more verbose version of the above tree, represented in XML format.

Shallow Logical Form

The parse trees are translated into a shallow logical form as defined by Hobbs in (Hobbs 1985; 1998). The sentence ‘The heart is a pump.’ has the shallow logical form:

be’(e0,x0,x1) & heart-nn’(e2,x0) & pump-nn’(e1,x1)

The variables e0, e1 and e2 reify the “be” relation and the properties of being a heart and being a pump, respectively.

LFToolkit (Rathod & Hobbs 2005) is used to convert parse tree outputs to a shallow logical form. LFToolkit works by generating logical form fragments corresponding to lexical items in the sentence and using syntactic composition relations to identify variables among logical form fragments. In this step, certain additional logical representation forms may be introduced, such as, for plural nouns, sets with variables that represent the individuals as well as the set implicit in the plural noun itself.

Transformations and Mapping

The mapping from this logical form fragments to the triple representation required by KM is performed by Mini-Tacitus (Hobbs *et al.* 1993). This step involves fragment combination, integration, and some reformulation, as well as, occasionally, the introduction of additional supporting or inferentially derived material. It is performed by backchaining and making assumptions where necessary, using a knowledge base of hand crafted axioms. In a sense, the KM system formulation provides the best (abductive) explanation of the content of the sentence.

Example

The output produced by NL is a set of triples. KM requires this form, stipulating that the triples’ slot names are defined in the KM component library as relations. That is, the lexicosemantics of each type of event and process (typically present in text as verbs) is represented by a set of relations (slots, defined in the domain ontology) for arguments, such as object-of and agent-of. Input defining new types results in the assertion of new concepts to the domain ontology, and input expressing a specific state of the world is recorded as new instantial information.

The sentence:

Oxygenated blood returns to the heart.

is parsed and translated into the shallow logical form:

oxygenate-vb’(e5,x2,x0) & blood-nn’(e2,x0) & return-vb’(e0,x0) & to’(e1,e0,x1) & heart-nn’(e4,x1)

Mini-Tacitus and a final post processor that reformats the output then produce the following set of triples:

```
(( e0 instance-of return )
( x0 instance-of blood )
( x0 object-of e0 )
( x0 object-of e5 )
( x4 instance-of heart )
( x4 destination-of e0 )
( e5 instance-of oxygenate )
( x2 agent-of e5 )
( e20 eventuality-of to ))
```

The relation ‘instance-of’ indicates eventualities corresponding to verbs and arguments of single argument predicates.

KM matches the triples with models of devices it already knows about and uses these to construct a model of the new device, in this case, the heart. Procedures developed for the Halo project were then used for answering questions.

Discovering Semantic Linkages

Introduction

KM is not robust to the errors and other shortcomings in the logical form. For example, the failure of NL to link together, using co-indexed variables, the structures in different parts of a sentence caused KM significant difficulties. KM lost a lot of information in these cases and sometimes even failed to recognize simple actions, events or instances. For example, the system produced the following logical form for the sentence “The heart is a pump that supplies blood to various parts of the body.”:

heart-nn’(x0) & be’(x0,x1) & pump-nn’(x2) & supply-vb’(x4,x3) & blood-nn’(x7) & to’(x6,x9) & various-adj’(x9) & part-nn’(x9) & of’(x8,x10) & body-nn’(x10)

Because of an incorrect parse, this logical form does not capture the facts that it is the heart that is supplying blood and that this supplying is to parts of the body. Telling only that there is a supply event and omitting what is supplying what, the final set of triples created is:

```
(( x0-heart instance-of heart )
( x0-heart is x1 )
( e0-is eventuality-of is )
( x2-pump instance-of pump )
( x4 supply x3 )
( x7-blood instance-of blood )
( x9-part destination-of x6 )
( x9-part MOD various )
( x9-part instance-of part )
( e8-of eventuality-of of )
( x10-body instance-of body )
( x8 of x10-body ))
```

Of these 12 triples, the then current version of the KM system could identify only 2, and created no unique model of the concepts. This was primarily because the fragmented nature of the logical forms left gaps in the context of the sentence; for example, the verb "supply" does not have an agent or an object. Unfortunately, this problem is pervasive in processing sentences with a high degree of syntactic complexity, as most sentences in scientific prose are.

To overcome these difficulties, we use abductive inference. The technique described in the following sections helped improve such partial output generated by LFToolkit. For the above example, the following output was generated using abduction:

((x4-heart instance-of heart)
(x4-heart of-type device)
(x4-heart is x1)
(e0-is eventuality-of is)
(x2-pump instance-of pump)
(e4-supply instance-of supply)
(x4-heart agent-of e4-supply)
(x3-blood object-of e4-supply)
(x3-blood instance-of blood)
(x3-blood of-type fluid)
(x9-part destination-of x6)
(x9-part MOD various)
(x9-part instance-of part)
(e8-of eventuality-of of)
(x8 of x10-body)
(x10-body instance-of body))

Highlighted triples indicate the discovered linkages.

With these inputs, that version of KM could not only identify 7 of the mentioned concepts, but also created 2 new models of the concepts, "heart" and "supply", a significant improvement as a result of simple abductive techniques.

Brief Overview of Mini-Tacitus

We have developed the Mini-Tacitus (Hobbs *et al.* 1993) for abductive inference on the interpretation of natural language discourse. It attempts to find the lowest cost proof of the shallow logical forms of the sentences in the text. It does so by backchaining over hand-crafted axioms in the knowledgebase, making assumptions where necessary (at a cost) and unifying or factoring propositions where possible, in order to get a more economical proof.

Using Abduction for Semantic Linkage

In complex sentences of the sort that occur in scientific text, syntactic analysis often fails because of bad parses to find the right predicate-argument relations. Essentially, variables from predications resulting from different lexical items are not integrated as they should be. In addition, there may be no syntactic evidence of this identity of arguments; the two words may be syntactically unrelated. An obvious case of this is where the words occur in different clauses, but it can also occur in syntactically complex single clauses, or even in such simple cases as nominal compounds. In "blood supply", there is no syntactic reason that the implicit relation between "blood" and "supply" should be the predicate-argument relation. We have developed a technique for capturing these semantic relations.

Revisiting previous sections, the sentence

The heart is a pump that supplies blood to various parts of the body.

produces the logical form

heart-nn'(x0) & be'(x0,x1) & pump-nn'(x2) & supply-vb'(x4,x3) & blood-nn'(x7) & to'(x6,x9) & various-adj'(x9) & part-nn'(x9) & of'(x8,x10) & body-nn'(x10)

The sparse linkage between individual predicates results either from the shortcomings of the syntactic parser, or from missing translations in LFToolkit. To recover this kind of information, we added axioms for all the domain-relevant types of entities and relations. These axioms are used as part of the knowledge base of Mini-Tacitus for performing abductive inferencing.

For example, two of the entities in the biology domain, mentioned in this sentence, are 'heart' and 'blood'. The axioms generated for these entities are

device'(e2,x1) & heart-nn'(e1,x1) ⟶ heart-nn'(e1,x1)
fluid'(e2,x1) & blood-nn'(e1,x1) ⟶ blood-nn'(e1,x1)

These axioms are a method of introducing constraints on arguments and supertype relations into a system that performs only backchaining. The first says that if something is a device and a heart, it is a heart. This may seem a counter intuitive procedure, but is very similar to the use of *et cetera* propositions discussed in (Hobbs *et al.* 1993). There, an axiom written

device'(e2,x1) & etc'(e1,x1) ⟶ heart-nn'(e1,x1)

could be paraphrased "being a heart is one way of being a device". Here the proposition heart-nn'(e1,x1) is playing the role of the et cetera proposition. A special mechanism in Mini-Tacitus blocks iterative backchaining on this rule.

Axioms were also created for relations, enabling backchaining from the relation to properties of possible agents and objects. Consider 'supply', for which the axiom is

device'(e2,x1) & fluid'(e2,x0) & supply-vb'(e3,x1,x0)
⟶ supply-vb'(e3,x1,x0)

According to this axiom, given a 'supply' event , it is assumed that there is a device 'x1' which supplies 'x0' which is a fluid. Mini-Tacitus then applies these axioms by backchaining on the entities and relations. It unifies similar predicates, when occurring in a single interpretation. The working of this unification or factoring is shown in figure 1.

No.	Axioms
	Relations
1	device’(e2,x1) & fluid’(e3,x2) & pump-vb’(e1,x1,x2) ⟶ pump-vb’(e1,x1,x2)
2	device’(e2,x1) & fluid’(e3,x2) & drain-vb’(e1,x1,x2) ⟶ drain-vb’(e1,x1,x2)
3	device’(e2,x1) & fluid’(e3,x2) & carry-vb’(e1,x1,x2) ⟶ carry-vb’(e1,x1,x2)
4	fluid’(e2,x1) & chemical’(e3,x2) & carry-vb’(e1,x1,x2) ⟶ carry-vb’(e1,x1,x2)
5	device’(e2,x1) & chemical’(e3,x0) & supply-vb’(e1,x1,x0) ⟶ supply-vb’(e1,x1,x0)
6	device’(e2,x1) & fluid’(e3,x0) &supply-vb’(e1,x1,x0) ⟶ supply-vb’(e1,x1,x0)
7	fluid’(e2,x1) & chemical’(e3,x0) & supply-vb’(e1,x1,x0) ⟶ supply-vb’(e1,x1,x0)
	Instances
8	device’(e2,x1) & heart-nn’(e1,x1) ⟶ heart-nn’(e1,x1)
9	fluid’(e2,x1) & blood-nn’(e1,x1) ⟶ blood-nn’(e1,x1)
10	device’(e2,x1) & atrium-nn’(e1,x1) ⟶ atrium-nn’(e1,x1)
11	chemical’(e2,x1) & oxygen-nn’(e1,x1) ⟶ oxygen-nn’(e1,x1)

Table 2: Axioms

The predicates “device’(e4,x1)” and “device’(e6,x2)” are unified to a single predicate “device’(e6,x2)”. As a result the unification leads to unification of all the instances of ‘x1’ with ‘x2’. Similarly, the multiple instances of ‘fluid’ are unified to deliver the following final logical form:

heart-nn’(e1,x2) & device’(e6,x2) & supply-vb’(e8,x2,x3) & fluid’(e7,x3) & blood-nn’(e3,x3)

This is in fact the actual information conveyed in the sentence, and is now properly reflected in the logical form.

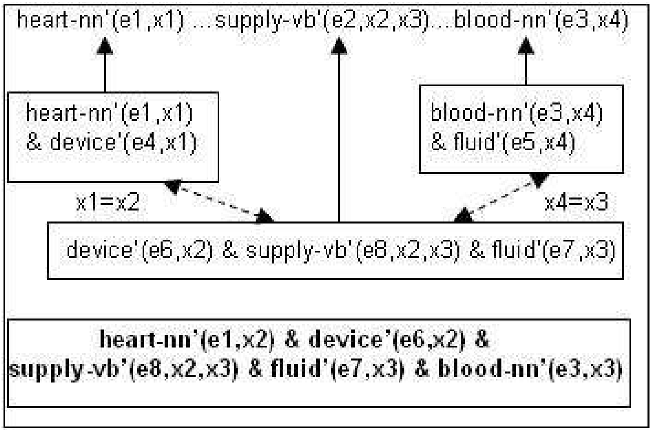
heart-nn'(e1,x1) ...supply-vb'(e2,x2,x3)...blood-nn'(e3,x4)
heart-nn'(e1,x1)
& device'(e4,x1)
blood-nn'(e3,x4)
& fluid'(e5,x4)
x1=x2
x4=x3
device'(e6,x2) & supply-vb'(e8,x2,x3) & fluid'(e7,x3)
heart-nn'(e1,x2) & device'(e6,x2) &
supply-vb'(e8,x2,x3) & fluid'(e7,x3) & blood-nn'(e3,x3)

Figure 1: Interpretation generation by Unification

Collection of Semantic Knowledge

Though we had demonstrated the feasibility of the idea of improving inadequate parse output using abductive inference, we still had to manually create the inference axioms, tailored to each situation. For general utility it is imperative to encode in the knowledge base a large set of axioms for domain-relevant entities and relations. This required that we determine somehow the kinds of information one might encounter in a paragraph about the heart, and from this create the appropriate axioms.

For this purpose, we scanned 10GB of webscraped material for biology texts related to the heart. In order to facilitate relevance of the information to be encoded in the axioms, we selected sentences in which the key concepts (“heart” and the appropriate other term being related by the axiom to be built) were separated by 7 words or fewer. Though most of these sentences could be rejected as not containing any relevant information, as in “... **heart** disease and high **blood** pressure...”, a total of 118 usable sentences were shortlisted from the dataset, and axioms incorporating 35 new verbs were added into the axiom knowledge base of Mini-Tacitus. Samples of axioms developed for relations and instances are provided in table 2.

Sometimes, axioms may be over-eager and unify too aggressively. When the text mentions multiple instances of devices or fluids, their incorrect unification has to be avoided. For example, we do not want to unify a heart and a lung just because they are both devices. A heuristic to avoid most such identifications simply blocks unification when the two entities are described by distinct nouns.

Disambiguation of Interpretations

Expansion of the axiomatic knowledge base often leads to ambiguities in interpretation. Simple relation words such as ‘supply’ may have various senses that imply different interpretations; for example, a device can supply a fluid, a fluid can supply a chemical, or a device can supply a chemical. Mini-Tacitus can deal effectively with such multiple interpretations by selecting the most probable cost-effective solution from the set of possible solutions. Referring back to the sentence

The heart is a pump that supplies blood to various parts of the body.

and the corresponding logical form

heart-nn’(x0) & be’(x0,x1) & pump-nn’(x2) & supply-vb’(x4,x3) & blood-nn’(x7) & to’(x6,x9) & various-adj’(x9) & part-nn’(x9) & of’(x8,x10) & body-nn’(x10) there are 3 candidate axioms for the relation ‘supply’ in table 2. Mini-Tacitus applies each of the axioms, generating multiple interpretations.

The application of axiom(5) generate:
device’(e2,x0) & heart-nn’(e1,x0) & ..... & chemical’(e3,x5) & supply-vb’(e4,x0,x5) & ..... & fluid’(e6,x7) & blood-nn’(e7,x7)

Factoring of the ‘device’ predicate from ‘heart-nn’ and ‘supply-vb’ yields a common ‘device’ predicate which is the agent of the supplying action. However, the object of ‘supply’ is assumed to be ‘chemical’ which does not have any backing from the logical form of the sentence. Similarly, the interpretation generation by application of axiom(7) produces:

device’(e2,x0) & heart-nn’(e1,x0) & ..... & chemical’(e3,x5) & supply-vb’(e4,x7,x5) & ..... & fluid’(e6,x7) & blood-nn’(e7,x7)

Assuming the cost of each proposition to be constant, the above two interpretations each have a cost of 6, whereas the interpretation obtained by application of axiom(6) is 5. Hence Mini-Tacitus selects the latter interpretation as the best solution.

Performance Evaluation

During its development, the system was developed using eight selected sentences. Appropriate rules were developed for perfect execution of the 8 sentences. The result was that 8 out of 8 sets of triples produced by NL were matched with the KM ontology and for 7 out of 8, models were created from the matched triples.

We then proceeded to systematically broaden the number of texts used. Next, 2 new paragraphs of 17 sentences were processed with the existing system, but with minimal human intervention. Specifically for NL, about five LFToolkit rules were added and about ten rules were added for labeling the arguments of previously unseen verbs. The output obtained showed that 16 out of 17 sets of triples produced by NL were matched with the KM ontology, and 9 models were created from the matched triples.

Third, we processed a paragraph of 10 completely new sentences without any human intervention or addition of new rules. The result was that the triples were matched with the KM ontology in 5 out of 10 cases and a model structure was generated in 2 of these cases. Most of the errors could be attributed to shortcomings in the LFToolkit rules so far implemented and to a lack of lexical knowledge. These results were encouraging; they indicate that beginning with sufficient lexical coverage and a more-general set of LFToolkit rules gives one a chance of reasonably robust performance in the face of unseen text.

The final system was evaluated by measuring the total number of concepts and relations captured by the KB from the NL outputs. A comparative study was performed on 22 sentences of a previously unseen text. The results appear in table 3. This comparison was based on the performance of the KRR system on the NL outputs before and after defining of the semantic linkages per sentence. A total of 35 new verbs were introduced as axioms.

	NL		KRR	
	Before	**After**	Before	**After**
S1	22	**24**	15	**17**
S2	41	**43**	32	**31**
S3	25	**26**	12	**13**
S4	26	**28**	15	**18**
S5	17	**19**	11	**13**
S6	22	**23**	15	**16**
S7	26	**28**	15	**18**
S8	18	**19**	8	**8**
S9	14	**15**	10	**11**
S10	12	**13**	7	**8**
S11	42	**44**	28	**30**
S12	10	**11**	5	**6**
S13	18	**20**	16	**15**
S14	22	**25**	9	**12**
S15	12	**13**	6	**7**
S16	19	**21**	12	**14**
S17	6	**7**	3	**4**
S18	19	**21**	12	**15**
S19	26	**28**	17	**17**
S20	21	**25**	15	**18**
S21	25	**28**	15	**19**
S22	22	**25**	9	**12**
Total	490	**552**	304	**350**

Table 3: Evaluation of performance

The result of this evaluation showed a 13% increase in the number of NL triples created which led to a 15% increase in the number triples that KM accepted from NL. These results demonstrate that the techniques described here hold some promise.

Conclusion

In this paper we have described the natural language component of a system we have built for learning models of devices from scientific text. Because of the syntactic complexity of scientific text, we cannot count on parses being entirely correct. In order to exploit all the information possible, in these cases we translate fragments of the sentence into independent fragments of logical form. Then we use the semantic properties of entities and relations, along with an abductive inference engine that minimizes the cost of interpretations by unifying propositions where possible, to recover the missing linkages between the independent fragments of logical form. Our experiments, evaluated over a series of trials, indicate that using abduction to overcome shortcomings in the parser and ‘mend’ inadequate parses is a promising approach to some of the difficulties that arise out of the complexity of scientific discourse.

References

Clark, P.; Harrison, P.; and Thompson, J. 2003. A knowledge-driven approach to text meaning processing. In *Proceedings of the HLT-NAACL 2003 workshop on Text meaning - Volume 9*, 1 – 6.

Friedland, N., and Allen, P. 2004. Project HALO: Towards a Digital Aristotle. In *AI Magazine*.

Hermjakob, U., and Mooney, R. J. 1997. Learning Parse and Translation Decisions From Examples With Rich context. In *Proceedings of the Association for Computational Linguistics(ACL)*.

Hobbs, J.; Stickel, M.; Appelt, D.; and Martin, P. 1993. Interpretation as Abduction. In *Artificial Intelligence Vol. 63, Nos. 1-2, pp. 69-142*.

Hobbs, J. R. 1985. Ontological Promiscuity. In *Proceedings, 23rd Annual Meeting of the Association for Computational Linguistics*, 61–69.

Hobbs, J. 1998. The Logical Notation: Ontological Promiscuity. In *Discourse and Inference: Magnum Opus in Progress*.

Mulkar, R.; Hobbs, J.; Hovy, E.; Chalupsky, H.; and Lin, C.-Y. 2007. Learning by Reading: Two Experiments. In *Proceedings of IJCAI 2007 workshop on Knowledge and Reasoning for Answering Questions*.

Rathod, N., and Hobbs, J. 2005. LFTookit. In *http://www.isi.edu/ñrathod /wne /LFToolkit /index.html*.