Learning by Reading: Two Experiments

Rutu Mulkar[1], **Jerry R. Hobbs**[1], **Eduard Hovy**[1], **Hans Chalupsky**[1], **Chin-Yew Lin** [2]

[1]Information Sciences Institute , University of Southern California

$\{rutu, hobbs, hovy, hans\}$ @isi.edu

[2]Microsoft Research Asia

$\{cyl\}$@microsoft.com

Abstract

This paper addresses the challenge of learning information by reading natural language text. The major aim is to map natural language input into logical expressions anchored upon concise and specific theories underlying the domains, in such a way that a reasoning engine can be used to answer questions about the input. We define a 3-step procedure, including parsing and abduction, and explore different implementations for the steps. Experiments were conducted in two domains, chemistry and biology, and the versatility of the approach suggests that extension to other domains is possible when the underlying theories are suitably specified.

1 Introduction

In this paper we address the problem of deriving semantic representations automatically from natural language text. This longstanding dream of AI has recently been revived in the Halo Project [Friedland and Allen, 2004], which investigated various existing Knowledge Representation and Reasoning (KRR) techniques and descried their potential and drawbacks.

Since part of the Halo systems' reading process was performed manually, we decided to investigate the feasibility of automating the entire process, from natural language sentences as input to answers to questions (that may involve reasoning) as output, and to test the implementation(s) in several domains. This paper is a brief early report of some of our work.

We modularized the process into a series of steps, namely NL parsing (Section 2.1), conversion to shallow logical form (Section 2.2), and abductive mapping to deeper form(s) (Section 2.3). To ingest and reason with the results, we employed two different KRR systems, with underlying models built at different degrees of completeness (Sections 3.1 and 3.2). The result is an automated flow from natural language text to a question-answering ability entirely independent of human intervention. The quality of the learning performed by the system has been preliminarily evaluated by asking questions regarding the textual input and measuring the amount of information automatically learned from natural language data.

We applied the system to two domains:

Chemistry: Two subsections of a high school chemistry textbook

Biology: Various paragraph-length texts describing the human heart.

2 Language Processing Steps

2.1 Parsing

In the chemistry domain, the first step was performed by the Charniak parser [Charniak, 2000]. In the biology domain, we used the CONTEX parser [Hermjakob and Mooney, 1997]. In both cases, the parse tree was converted into a shallow logical form as explained in Section 2.2.

A example parse tree produced by CONTEX for the sentence "the heart is a pump" is:

```
(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]
```

Since we were not in this project focusing on parsing itself, we avoided parser problems by manually simplifying sentences' syntactic structure prior to parsing where necessary in the following ways:

- Splitting sentences into two if joined by a conjunction.

- Removing appositives (described by bracketed NPs adjoining an NP) and writing it as a separate sentence.

- Replacing within-sentence images representing chemical formulae by the chemical formulae expressed in plain text.

2.2 Shallow Logical Form

In the chemistry domain, the parse tree was first converted into a series of minimal triples called Basic Elements (BEs) [Hovy *et al.*, 2005], defined as triplets of words consisting of a head and a modifier or argument, with its relation to the head, and then into a shallow logical form (LF) [Hobbs, 1985; 1998], defined as a list of conjoined expressions with linked variables.

A sample analysis into BEs for the sentence :

"Citric acid in lemon juice has a sour taste" is

$$< citric_{JJ}|NN - JJ|acid_{NN} > NP$$
$$< acid_{NN}|NN - NN_{IN}|juice_{NN} > NP$$
$$< lemon_{NNP}|NN - NNP|juice_{NN} > NP$$
$$< acid_{NN}|has_{AUX}|ARG0 > VP$$
$$< ARG1|has_{AUX}|taste_{NN} > VP$$
$$< a_{DT}|NN - DT|taste_{NN} > NP$$
$$< sour_{JJ}|NN - JJ|taste_{NN} > NP$$

In the chemistry domain, a simple script was developed to map the BEs into LF expressions.

In the biology domain, the CONTEX parse tree was converted directly into LF using LF Toolkit [Rathod and Hobbs, 2005], whose rules traverse the parse tree and output an LF expression for each appropriate word and each syntactic branch node, using composition relations to identify variables among logical form fragments.

For example the sentence "The heart is a pump" has the shallow logical form:

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

where the variables *x0* and *x1* represent the heart and the pump respectively and the variables *e0, e1* and *e2* reify the "be" relation and its components (the properties of being a heart and being a pump, respectively).

2.3 Transformations to Deeper Semantic Form

The LF shallow representations are not sufficiently 'semantic' to support significant reasoning. In particular, sentences that express rules have to be converted into axiom format, determiners have to be converted into the appropriate referential expressions, verb arguments have to be provided with explicit relation names, etc.

Since these transformations may influence one another, and since they are in some cases not deterministic but depend on (usually correct) assumptions, we employ the abductive reasoner Mini-Tacitus [Hobbs *et al.*, 1993] to perform them. In a sense, the resulting formulation provides the best (abductive) explanation of the content of the sentence. Axioms were crafted manually to allow the system to backchain from the shallow logical form to a form that could be used by the KRR system. In the biology domain, for example, the KRR system required both part-of-speech information as well as verb argument names from its component library relations. The following example shows an input sentence, its shallow logical form, and the final output after transformation.

Oxygenated blood returns to the heart.

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

((x2 agent-of e5)
(x0 object-of e0)

(x4 instance-of heart)
(heart pos noun)
(x0 instance-of blood)
(blood pos noun)
(x0 object-of e5)
(x4 destination-of e0)
(e5 instance-of oxygenate)
(oxygenate pos verb)
(e20 eventuality-of to)
(e0 to x0)
(e0 instance-of return)
(return pos verb))

Note the insertion of verb argument names such as *object-of, destination-of, agent-of* etc. The relation *instance-of* connects arguments to their model types.

3 Knowledge Representation and Reasoning

For the chemistry domain, we employed the KRR system PowerLoom [Chalupsky *et al.*, 2006], built at ISI, for which first-order logical axioms had to be created manually at ISI. In the biology domain, we employed the KRR system Knowledge Machine (KM) [Clark *et al.*, 2003], built at the University of Texas in Austin, using models of the domain built by our collaborators in Texas. The details are explained in Sections 3.1 and 3.2.

3.1 Chemistry Domain

The corpus for the chemistry domain was a high school chemistry textbook. We tested the system with two selected subsections of the textbook, a total of 133 sentences, concerning acids and bases.

Technical Details

The input to PowerLoom was first-order logic axioms, represented in the Knowledge Interchange Format (KIF) [Genesereth, 1991]. Three types of knowledge were captured.

General Facts: Because we were processing a textbook, it is reasonable to resolve what in isolation would be a generic-specific ambiguity in favor of the generic interpretation. This allowed many sentences to be converted into axioms in which the subject implies the predicate. Frequently, for example, the subject is a chemical term and the predicate defines this term, as in

An H+ ion is a proton.
(FORALL (?e2 ?x1 ?e4 ?x2 ?e3)
(⇒(AND (nn ?e4 ?x2 ?x1)
(h+ ?e3 ?x2)
(ion ?e2 ?x1))
(EXISTS (?e8 ?e6 ?z1 ?x3)
(AND (be ?e8 ?e2 ?e6 ?z1)
(proton ?e6 ?x3)
))))

That is, if something is an ion bearing some underspecified relation "nn" to H+, then it is a proton.

Causal Facts: The presence of such causal keywords as "because", "when", and "implies" licenses the extraction and formulation as axioms of causal rules, as in

When bases are added to acids, they lower the amount of acid.

```
(FORALL (?e3 ?z1 ?x1 ?x2 ?e4
    ?e5 ?s1 ?e10 ?e6 ?e7 ?e9 ?s2)
  (⇒ (AND (add ?e3 ?z1 ?x1 ?x2)
        (base ?e4 ?x1)
        (plural ?e5 ?x1 ?s1)
        (they ?e10 ?x1)
        (to ?e6 ?e3 ?x2)
        (acid ?e7 ?x2)
        (plural ?e9 ?x2 ?s2))
    (EXISTS (?e2 ?x4 ?e13 ?e14 ?x5 ?e15 ?e13)
      (AND (when ?e13 ?e3 ?e2)
        (lower ?e2 ?x1 ?x4)
        (amount ?e13 ?x4)
        (of ?e14 ?x4 ?x5)
        (acid ?e15 ?x5)))))))
```

This axiom captures the relation between the "add" and "lower" in the sentence through "when". When the adding event *e3* is performed, the lowering event *e2* occurs.

Reaction Theory: In sentences involving chemical reactions, predicates such as "dissociate" in the shallow logical form can be mapped to an underlying theory of reactions, as in

NaOH dissociates into Na+ and OH- ions when it dissolves in water.

```
(FORALL (?e218 ?e1 ?e217 ?e216 ?e17 ?e4
    ?e0 ?x5 ?e8 ?x1)
  (⇒ (AND (REACTION ?e218 ?e1 ?e217)
        (in ?e216 ?e8 ?e217)
        (when ?e17 ?e218 ?e8)
        (into ?e4 ?e218 ?e0)
        (water ?e217 ?x5)
        (naoh ?e1 ?x1))
    (EXISTS (?e5 ?e10 ?x3 ?x2 ?e11 ?e16 ?s1)
      (AND (FORMS ?e8 ?e1 ?e0)
        (into ?e4 ?e218 ?e0)
        (and ?e0 ?e5 ?e10)
        (ION ?e5 ?x3 na+)
        (ION ?e10 ?x2 oh-)
        (ion ?e11 ?x2)
        (plural ?e16 ?x2 ?s1)))))))
```

Here the words "dissociate" and "dissolve" are mapped into the core theory concepts REACTION and FORMS.

Analysis

The correct logical forms were generated for 91 out of the 133 sentences. Among the causes of errors were parse errors generated by the Charniak parser, errors due to incorrect linking of modifiers with the syntactic head in the BEs, and bugs in the conversion from BEs to logical form.

The compatibility of the NL and KRR systems can best be judged by the degree to which the latter can reason with data generated by the former. PowerLoom was able to perform certain transitivity inferences and also answer *what* and *how* questions.

Knowledge from NL:
H3O+ is the conjugate acid of H2O.
Acids cause certain dyes to change color.
Bases have a bitter taste and feel slippery.
Soap is a base.

Questions
Question(T/F): H3O+ causes certain dyes to change color.
Answer: True
Question (what): Soap has WHAT taste?
Answer: 1: ?what=taste 2: ?what=bitter
Question (how): Soap feels how?
Answer: 1: ?how=slippery

Such examples show that the system is indeed learning from the text by being able to perform appropriate derivations in some cases.

Nevertheless, PowerLoom encountered several problems. The most significant problem was that natural language tends to be much more verbose than the theories typically handled by theorem provers. Also, there was a lack of connection between the knowledge that was extracted and a background theory to link and constrain the NL output to. PowerLoom could not ignore irrelevant information in the axioms derived from the input text, which caused explosion in forward chaining. Because of the *forall — there exists* structure of the axioms, this involved an explosion in the number of Skolem constants and functions generated.

3.2 Biology Domain

In the biology domain, several dozen paragraph-length texts describing the human heart were analyzed. Underlying domain models were built at the University of Texas at Austin and implemented in KM.

Technical Details

The KM system takes as input the triples produced by Mini-Tacitus and matches them with its pre-built models of devices. Combining elements from the input and the models, it constructs a model of the new device, in this case, the heart. When complete, inference procedures developed for the Halo project are used for answering questions. An example of the input required was given in Section 2.3.

Analysis

Unlike PowerLoom, KM ignores any information that it cannot process. As a result, verbose outputs from the NL component do not create problems. However, KM is not robust against errors in the logical form. For example, if NL fails to link the structures in different parts of a sentence appropriately, some information is lost during KM reasoning.

We developed rules for Mini-Tacitus to transform input into the appropriate logical form in several stages, testing the coverage of each stage on unseen, novel input for the next. The initial set of rules, developed for a set of 8 sentences, handled enough of the second batch of input (17 sentences) that only 15 new LF Toolkit rules had to be added (10 of them for labeling the arguments of new verbs). With these additions, almost all the triples produced were matched to the KM models, causing 9 models to be created.

For the third stage, a paragraph of 10 new sentences was read without any human intervention or addition of new rules. The triples for 5 of the 10 sentences were successfully matched with the KM models, resulting in a model structure in 2 of these cases.

These results are encouraging, in that most of the errors can be attributed to shortcomings in the LF Toolkit rules we have so far implemented, which amounts to the lack of syntactic and lexical knowledge. We are addressing these shortcomings in a systematic fashion.

4 Comparing KRR Systems and Their Knowledge Bases

We make the following observations when comparing PowerLoom and KM for the task of learning by reading:

- KM is more tolerant to natural language verbosity than PowerLoom, since it rejects unwanted triplets and works only with data it understands.

- PowerLoom is a pure reasoning system, without any pre-constructed models or ontologies. Its domain models have to be specially built, a task that requires non-trivial expertise. In contrast, KM has a standard ontology and set of models, built for the Halo project. While building and extending its models also require considerable expertise, a methodology is in place, together with a standard library of building blocks.

- Learning and QA are easier to understand and trace in PowerLoom than KM. In KM it is difficult to determine whether the answers derive entirely from the text or from the already present models. However it can be argued that models are only selected when KM receives sufficient backing from the NL output.

- Due to the built-in models in KM, and the explanation capabilities developed for Halo, the outputs it generates are more precise and robust than those of PowerLoom.

5 Summary and Future

In this paper, we have described two experiments in learning knowledge from textbooks. In the first, in the chemistry domain, we used a relatively knowledge-poor theorem prover, PowerLoom, and showed its utility in answering questions about the text requiring inferences. In the second experiment, in the biology domain, we have shown that interpreting textbook-like passages with respect to a rich set of models of devices, built in the KM system, can be used to create more complex models.

In both cases, we envision cycling from chapter to chapter, e.g., learning a theory of chemical reactions from Chapter 2 and using that to interpret and reason about the information on acids and bases in Chapter 5. A feedback loop in which the model built for the text so far would be used in interpreting and disambiguating subsequent sentences in the text, and in fact possibly produce specific requests to the NL engine to locate and read passages about specific topics, presents an interesting challenge for the future.

Both experiments indicate that NL and KR technologies have reached a point where learning by reading is a serious possibility.

References

[Chalupsky *et al.*, 2006] Hans Chalupsky, Robert M. Mac-Gregor, and Thomas A. Russ. Powerloom manual, University of Southern California. In *http://www. isi. edu/ isd/ LOOM/ Power- Loom/ documentation /manual.pdf*, 2006.

[Charniak, 2000] Eugene Charniak. A maximum-entropy-inspired parser. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, pages 132–139, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[Clark *et al.*, 2003] Peter Clark, Phil Harrison, and John Thompson. A knowledge-driven approach to text meaning processing. In *Proceedings of the HLT-NAACL 2003 workshop on Text meaning - Volume 9*, pages 1 – 6, 2003.

[Friedland and Allen, 2004] Noah Friedland and Paul Allen. Project halo: Towards a digital aristotle. In *AI Magazine*, 2004.

[Genesereth, 1991] M.R. Genesereth. Knowledge interchange format. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 599–600, San Mateo, CA, USA, April 1991. Morgan Kaufmann Publishers.

[Hermjakob and Mooney, 1997] Ulf Hermjakob and Raymond J Mooney. Learning parse and translation decisions from examples with rich context. In *Proceedings of the Association for Computational Linguistics(ACL)*, 1997.

[Hobbs *et al.*, 1993] Jerry Hobbs, Mark Stickel, Douglas Appelt, and Paul Martin. Interpretation as abduction. In *Artificial Intelligence Vol. 63, Nos. 1-2, pp. 69-142*, 1993.

[Hobbs, 1985] Jerry R. Hobbs. Ontological promiscuity. In *Proceedings, 23rd Annual Meeting of the Association for Computational Linguistics*, pages 61–69, 1985.

[Hobbs, 1998] Jerry Hobbs. The logical notation: Ontological promiscuity. In *Discourse and Inference: Magnum Opus in Progress*, 1998.

[Hovy *et al.*, 2005] Eduard Hovy, Chin-Yew Lin, and Liang Zhou. A be-based multi-document summarizer with sentence compression. In *Proceedings of Multilingual Summarization Evaluation (ACL 2005 workshop)*, 2005.

[Rathod and Hobbs, 2005] Nishit Rathod and Jerry Hobbs. Lftookit. In *http://www. isi.edu/ñrathod /wne /LFToolkit /index.html*, 2005.