30 June 2006

Reading to Learn

0002AC – Final Technical Report

Contract No.: HR0011-05-C-0073

SRI Project: 16551

Prepared by:

Principal Investigators:

Dr. David J. Israel
Tel: (650) 859-4254
Fax: (650) 859-3735
Email: israel@ai.sri.com

Subcontractor:

Mr. Peter E. Clark
The Boeing Company
Tel: (425) 865-4304
Fax: (425) 766-5217
Email: clarkp@puffin.rt.cs.boeing.com

SRI Administrative Contact:

Ms. Rachel Stahl
Tel: (650) 859-2004
Fax: (650) 859-6171
Email: Rachel.stahl@sri.com

SRI Financial Data Contact:

Kelli M. Connolly
Tel: (650) 859-2641
Fax: (650) 859-3735
Email: kconnolly@ai.sri.com

This material is based upon work supported by the
Defense Advanced Research Projects Agency
DARPA/IPTO
Learning to Read
Cognitive Information Processing Technology
ARPA Order No. T412
Program Code No. 4D10
Issued by DARPA/CMO under Contract # HR0011-05-C-0073

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	30 June 2006	Final (2/14/05 – 05/31/06)

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Reading to Learn	C – HR0011-05-C-0073 PR – SRI P16551
6. AUTHOR(S) David Israel, Peter Clark, Phil Harrison, John Thompson, Rick Wojcik, Tom Jenkins	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)		8. PERFORMING ORGANIZATION REPORT NUMBER
SRI International 333 Ravenswood Avenue Menlo Park, CA 94025	The Boeing Company P.O. Box 3707 Seattle, WA 98124	SRI P16551

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING / MONITORING AGENCY REPORT NUMBER
U. S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT	12 b. DISTRIBUTION CODE
Approved for public release; distribution unlimited.	

13. ABSTRACT (Maximum 200 words)

One of the most important methods by which human beings learn is by reading, a task that includes integrating what was read with existing, prior knowledge. While in its full generality, the reading task is still too difficult a capability to be implemented in a computer, significant (if partial) approaches to the task are now feasible. Our goal in this project was to study issues and develop solutions for this task by working with a reduced version of the problem, namely working with text written in a simplified version of English (a Controlled Language) rather than full natural language. Our experience and results reveal that even this reduced version of the task is still challenging, and we have uncovered several major insights into this challenge. We describe our work and analysis, present a synthesis and evaluation of our work, and make several recommendations for future work in this area. Our conclusion is that ultimately, to bridge the "knowledge gap", a pipelined approach is inappropriate, and that to address the knowledge requirements for good language understanding an iterative (bootstrapped) approach is the most promising way forward.

14. SUBJECT TERMS	15. NUMBER OF PAGES
	30
	16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Contents

List of Figures and Tables

1. Introduction

One of the most important methods by which human beings learn and increase their knowledge and understanding is by reading. Consider a human in a state in which she already knows something about some domain or area. She reads some material in that domain that contains knowledge that she does not yet have; her existing knowledge, including her knowledge of the language in which the material is presented, allows her to understand the material. Having read and understood the text, she incorporates newly acquired knowledge into her pre-existing knowledge; thereby enabling her to do new things, for example, answer questions that she couldn't answer before. As **reading to learn** is one central activity of humans, so too one central goal of AI is to create systems able to process natural language text into a machine understandable form so that the new content can be incorporated into existing knowledge bases and, thus, be rendered amenable to automatic reasoning methods, ultimately to enable question answering.

In its full and unrestricted generality, the reading task is still too difficult a capability to be implemented; however significant, if partial, approaches to the goal are now feasible. In particular, one can factor the reading task into two parts:

(i) language processing; and
(ii) Knowledge integration (interpretation and integration of the new knowledge into an existing knowledge base)

Based on this division, our **goal** in this project was to study issues in reading to learn, by working with a **reduced version** of the problem, namely working with **controlled language (CL) texts**, rather than unrestricted natural language (NL). This allowed us to side step some of the issues in full natural language processing (i), and concentrate on issues in machine understanding and knowledge integration (ii).

Our work has primarily been in the domain of chemistry, in which a prior, hand-built knowledge base already exists, created for Vulcan's Halo Pilot project (Barker et al, 2004). Specifically, we have focused on 6 pages of chemistry text concerning acid-base equilibrium reactions, namely pp614-619 of (Brown, LeMay and Bursten, 2003). Our methodology was as follows:

1. Rewrite the 6 pages of chemistry text into our controlled language, CPL
2. Extend and use our CPL interpreter to generate logic from this
3. Integrate this new knowledge with an existing chemistry knowledge base (from the Halo Pilot)
4. Assess the performance of the CPL-extended KB with the original
5. Report on the problems encountered and solutions developed

Our experience and results, which we present here, reveal that even this reduced version of the task is still challenging, and we have uncovered several major insights into this challenge. In particular, our work reveals the need for an iterative (bootstrapped) approach to reading rather than a traditional "waterfall" approach; for extensive use of background knowledge to guide interpretation; and a radical revision of traditional knowledge representation structures to support knowledge integration. We describe our work and analysis, present a synthesis and evaluation of our work, and describe several key recommendations for future work in this area. This work has turned out to be a fascinating and

exciting investigation into the challenges of the full reading to learn task, and we hope this report communicates this experience and motivates further research.

As a vehicle for this research, this project has also involved substantial technical development of our CPL (Computer-Processable Language) interpreter. CPL is described in (Clark et al., 2005) and in additional presentations available on request, and there is also a CPL Users Guide available which enumerates the full list of rules and advice messages for authoring in CPL (Thompson 2006). We will mention some of the features of CPL in this report where relevant, but not present extensive technical detail on CPL itself, in order to maintain focus of this report on the challenge of learning by reading.

2. Framework: The Knowledge Gap

There is a fundamental gap between real natural language text, on one hand, and an "ideal" logical representation of that text that integrates easily with pre-existing knowledge, on the other. Importantly, this gap arises from more than just grammatical complexity; it involves multiple other factors that we describe in this report. For full text comprehension, this gap must be bridged.

Our approach in this research was to reformulate the original target text into our controlled language, CPL ("Computer-Processable Language"). There are two ways this can be done, illustrated in Figure 1, both of which we have investigated:

1. Write CPL, which is "close" to the original English, i.e., is essentially a grammatical simplification of the original text with no/little new knowledge added. While in this project this reformulation is done by hand, one can plausibly imagine this reformulation could be performed automatically by some suitable software.

2. Write CPL, which fully captures the underlying knowledge that the author intended to convey, essentially treating CPL as a kind of declarative rule language. In this case, there is a significant gap between the original text and CPL reformulation, including significant new knowledge injected in the reformulation.

In formulation 1, while the CPL is "faithful" to the original English, the logical interpretation of the CPL retains much of the incompleteness and "messiness" of the text. As a result, it turns out to be difficult to support significant reasoning and inference with the final logic, and to integrate it with pre-existing knowledge. We describe this extensively in Section 3. Conversely, in formulation 2, although the resulting logic is clean and inference-supporting, the gap has only been bridged through significant manual intervention in authoring the CPL, unlikely to be performed automatically. We describe this extensively in Section 4. In both formulations the "gap" between the messiness of real language, and the tidiness required for formal reasoning, is an obstacle. In this report, we provide a detailed analysis of this gap, its causes, and recommendations for how to proceed to bridge it in future.

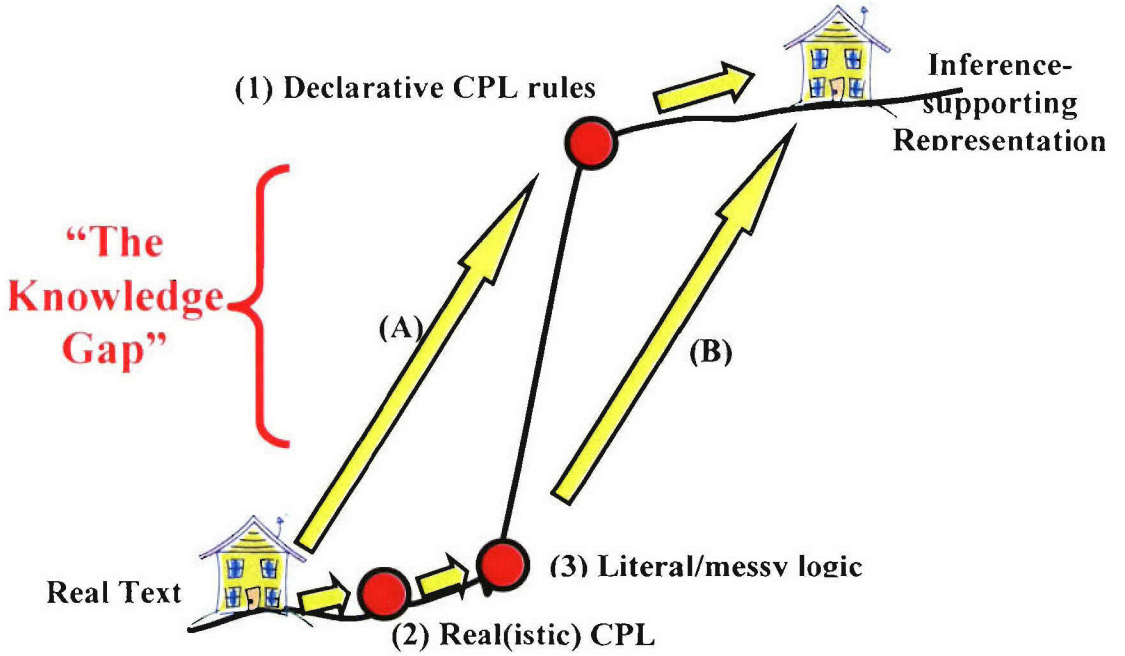


Figure 1: There is a significant knowledge gap between real language (the starting point, the house at the bottom of the cliff) and inference-supporting logic (our goal, the house at the top of the cliff). Writing the declarative rules underlying the text in CPL (1) crosses the gap (A), but only by virtue of significant and non-automatable manual intervention. Conversely, while CPL close to the original text (2) might plausibly be generated automatically, and logic generated from that (3), a significant gap (B) still remains between that logic and that required to support inference.

3. Analysis I: Sentence-by-Sentence Translation

3.1 Introduction

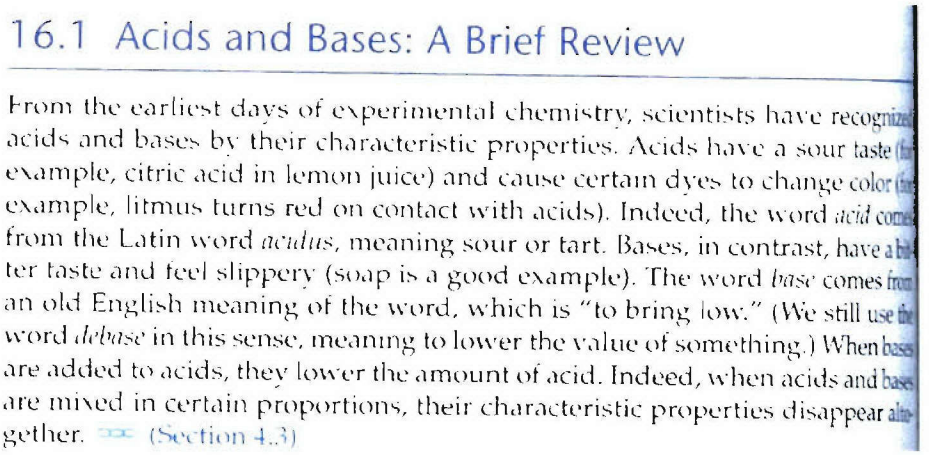
As Figure1 illustrates, there are two rather different approaches to bridging the gap. In the first approach, we have written CPL, which is reasonably close to the original English (bullet (2) in Figure 1), from which logic is then generated. However, as we shall describe, the resulting logic is "messy" (bullet (3) in Figure 1), essentially retaining much of the unwanted imprecision, overgenerality, and errors of the original text. These "problems" are things, which a human reader typically does not even notice, and almost unconsciously he/she fills in and corrects the information he/she is reading. However, for a computer, they pose serious problems. In addition, to support inference, the computer needs extensive background knowledge about what the words/predicates mean, knowledge that was often absent in our pre-existing background KB. Thus, although the result is syntactically logic, it is semantically "messy" and difficult to use.

In this Section, we describe the results of taking this path to logic from the text. The six pages of chemistry text were rephrased as approximately 280 CPL sentences, and logic generated from them. Although some knowledge and selectivity was injected into these reformulations, they are still largely faithful to the original English, and the reformulation process might plausibly be automated. We illustrate this process with two paragraphs in the textbook; the full list of 280 sentences and the corresponding logic is available on request. For expediency, some of the "fluff" in the text, e.g.,

historical notes, motivational anecdotes, were skipped during this encoding, although they could easily (if laboriously) also be encoded.

3.2 Case study 1: Paragraph 1

The six pages of text starts as follows:



16.1 Acids and Bases: A Brief Review

From the earliest days of experimental chemistry, scientists have recognized acids and bases by their characteristic properties. Acids have a sour taste (for example, citric acid in lemon juice) and cause certain dyes to change color (for example, litmus turns red on contact with acids). Indeed, the word *acid* comes from the Latin word *acidus*, meaning sour or tart. Bases, in contrast, have a bitter taste and feel slippery (soap is a good example). The word *base* comes from an old English meaning of the word, which is "to bring low." (We still use the word *debase* in this sense, meaning to lower the value of something.) When bases are added to acids, they lower the amount of acid. Indeed, when acids and bases are mixed in certain proportions, their characteristic properties disappear altogether. (Section 4.3)

This text illustrates many typical challenges that arise. Consider the first sentence

> "From the earliest days of experimental chemistry, scientists have recognized acids and bases by their characteristic properties."

To fully understand this requires already having some basic notion of time, chronologies, time periods, and their start and ends. It requires recognizing the idiom-like phrase "earliest days" as meaning "the start of". The sentence also includes generic references to scientists, acids, bases, and properties, and the challenge of interpreting generics (e.g., does it mean that all scientists recognize all acids all the time?). It includes a vague reference to "characteristic properties" -- which properties, exactly, are being referred to there? Or how does this vague notion get recognized in the KB? Similarly, what sense of the verb "recognize" is intended here? This is particularly challenging as the author is not referring to specific recognition events, rather is referring to the state of understanding of scientists in the past and present. Later sentences in the paragraph require prior knowledge about words and meaning, i.e., prior knowledge that there exist symbol systems (e.g., languages) used to describe the world.

In this particular paragraph, we have skipped much of this text, as it is not central to the chemistry knowledge we are interested in. The CPL encoding we wrote looks as follows:

```
Acids have a sour taste.
Acids cause some dyes to change color.
Bases have a bitter taste.
Bases have a slippery feel.
```

The logic generated from these four sentences looks as follows:

```
;;; Acids have a sour taste.
FORALL ?acid
  isa(?acid, Acid)
==>
```

```
EXISTS ?taste
  isa(?taste, Taste-Value)
  taste(?acid, ?taste)
  value(?taste, *sour)

----------

;;; Acids cause some dyes to change color.
FORALL ?acid
  isa(?acid, Acid)
==>
EXISTS ?dye, ?color, ?change:
  isa(?color, Color-Value)
  isa(?dye, Substance)
  causes(?acid, isa(?change, Reaction) AND object(?change,?color)
                  AND raw-material(?change,?dye))

----------

;;; Bases have a bitter taste.
FORALL ?base:
  isa(?base, Base)
===>
EXISTS ?taste:
  isa(?taste, Taste-Value)
  taste(?base, ?taste)
  value(?taste, *bitter)

----------

;;; Bases have a slippery feel.
FORALL ?Base:
  isa(?Base, Base)
===>
EXISTS ?feel:
  isa(?feel, Sense)
  possesses(?base, ?feel)
  property(?feel, *slippery)
```

The CPL interpreter (in this application) is using the ontology from the Halo KB as its target ontology. The ontology contains approximately 3000 concepts and 400 relations (predicates), a subset of these being directly related to chemistry. A table provides a mapping from words to concepts, and the CPL interpreter also makes use of WordNet to handle words, which are not directly in this table (by climbing the hypernym tree from the user's word until a word which is in the table is encountered). Thus, in some cases, the concept (word sense) found for a user's word is more general than that directly given by the user.

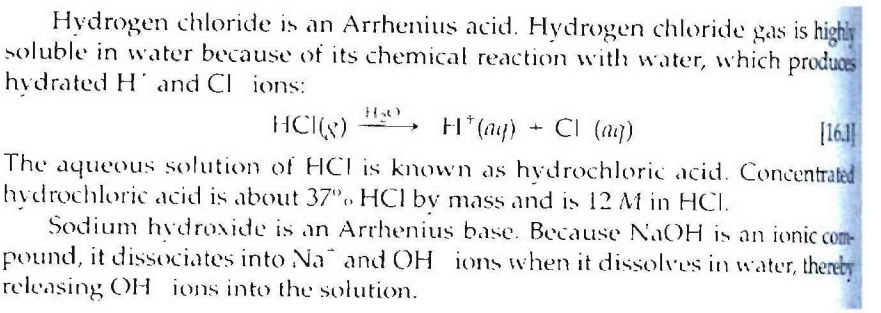
In some cases, the generated logic is sensible, e.g., for "Acids have a sour taste", as the notions of taste and sour are known. However, in other cases the logic is not sensible, for a variety of reasons. An interesting case is the second sentence "Acids cause some dyes to change color." Taken literally, the sentence is ambiguous, over-general, and erroneous:

- **Metonymy:** Strictly (at least in the Halo KB), only events can cause things, not objects. The sentence is referring to some (unstated) event involving acids that causes the change, and the word "acid" can be viewed as a metonymic reference to some event like "adding acid"

- **Presupposition:** The sentence omits (presupposes) contextual knowledge about how this change can take place, for example: The acid is in contact with the dye, the dye is not already the changed color, etc.
- **Ambiguity:** The sentence is ambiguous about whether the changing is a one-off or continuously ongoing event
- **Complex semantics:** The phase "some dyes" really means "all instances of some types of dyes"; that is, it assumes prior knowledge that there is a natural grouping of dyes into types, and that each type is characterized by whether all its members change color with acids, or whether they all do not.

As a result, the logic representing the author's intended meaning would be substantially more complex and different than the "literal" logic produced by the CPL interpreter. Moreover, this logic would include additional knowledge not present in the original text (e.g., that the acid and dye must be touching); thus, it is in principle infeasible to generate this logic from the sentence alone - rather, substantial background knowledge is also needed (either preprogrammed or itself acquired through some bootstrapped learning process). Given all this, the logic that we have generated, taking a mechanical translation process which does not handle these issues, is largely unusable for meaningful inference. As we will show later in our second analysis, we can alternatively create richer CPL which avoids these problems, and generates inference-supporting logic - but of course we have then manually crossed the gap which we wish the machine to ultimately be able to bridge.

3.3 Case Study 2: Paragraph 3



Hydrogen chloride is an Arrhenius acid. Hydrogen chloride gas is highly soluble in water because of its chemical reaction with water, which produces hydrated H^+ and Cl^- ions:

$$HCl(g) \xrightarrow{H_2O} H^+(aq) + Cl^-(aq) \quad [16.1]$$

The aqueous solution of HCl is known as hydrochloric acid. Concentrated hydrochloric acid is about 37% HCl by mass and is 12 *M* in HCl.

Sodium hydroxide is an Arrhenius base. Because NaOH is an ionic compound, it dissociates into Na^+ and OH^- ions when it dissolves in water, thereby releasing OH^- ions into the solution.

As a second case study, consider paragraph 3 of the text above. In this case, the original English is simpler and cleaner, and as a result the corresponding CPL (shown below) and final logic is also more usable. We include this example here to show that the task is not universally difficult - there do exist passages which are more amenable to machine processing, and where sensible knowledge can plausibly be extracted automatically. The CPL interpreter is able to recognize and represent physical quantities (e.g., 12 M), and the ontology includes the required chemical concepts (e.g., HCl, H-plus). This is not to say this paragraph is completely straightforward: in particular, representing explanations (proofs) described in the text (the two occurrences of "because") is challenging, and the CPL omits the clausal dependencies that the text presents. The CPL reformulation for this paragraph looks as follows:

```
Hydrogen chloride is an Arrhenius acid.
Hydrogen chloride gas is highly soluble in water.
Hydrogen chloride gas in water reacts with the water.
The reaction produces H-plus ions and Cl-minus ions.
HCl is hydrogen chloride.
```

```
Hydrochloric acid is an aqueous solution of HCl.
37 percent of the mass of concentrated hydrochloric acid is HCl.
The concentration of HCl in concentrated hydrochloric acid is 12 M
Sodium hydroxide is an Arrhenius base.
NaOH is sodium hydroxide.
NaOH is an ionic compound.
NaOH dissolves in water.
NaOH dissociates in water.
The dissociating produces Na-plus ions and OH-minus ions.
```

3.4 Synthesis

We have proceeded through the 6 pages of text in a similar manner, producing approximately 280 CPL sentences and corresponding logical clauses. In fact, most of the knowledge is somewhat peripheral to our original goal of answering AP questions with the interpreted knowledge. In the second analysis (Section 4), we present a more detailed study of the (few) parts of the study directly relevant to AP questions. This is not to say that the rest of the text is irrelevant, just not essential for passing the AP chemistry test.

Throughout the encoding process we encountered numerous encoding challenges; some of these are language related, while some are more semantic issues. Some are easier to resolve, while some represent major research challenges. We now present an overall summary of the main challenges we encountered, along with a rough assessment of their degree of difficulty that they pose for, say, a 5-year program targeted at the reading task (**green**=easy, **yellow**=medium, **red**=difficult, **black**=extremely difficult). "green" items are simply a matter of further software development for our CPL interpreter, while "red" and "black" also involve solving major research questions. The example sentences below are all from the original chemistry text.

1. **Idioms/special-purpose phrases** (yellow)

   "From the earliest days of experimental chemistry…"
   "The reaction favors transfer of…"
   "According to their definition…"

   Throughout the text, we encounter words and phrases ("favors transfer", "from the earliest days", "according to the definition") which have special-purpose meanings in the chemistry context, and where a literal interpretation of the language is largely meaningless. While one can envisage writing special-purpose software modules for handling these (e.g., an Earliest-Days processor), the challenge is simply the vast number of such phrases, which exist in language. Ideally, an advanced language system, with suitable background knowledge, might be able to guess the intended meaning of new phrases which are encountered based on the existing meaning of the words, combined with strong expectations about what sort of thing the author might be trying to say.

2. Interpreting **generics** (red)

   "Acids cause some dyes to change color."
   "A Bronsted-Lowry acid always reacts with a nearby Bronsted-Lowry base."

   Generic sentences are sentences about objects in general, rather than some specific individual(s) in the world. Generics are ubiquitous in tutorial texts. They are challenging to interpret because **quantifier scop**ing is often ambiguous; key **presuppositions** are often unstated but need to be identified; and **exceptions** almost always exist to the general rule (thus they rarely translate into a simple logical axiom). Our CPL interpreter assumes universal quantification over the main verb's

subject. It is not able to perform the complex task of identifying and inserting unstated presuppositions.

3. Handling **negation.** (green)
   "Some substances containing hydrogen are not acids."
   "The transfer leaves no undissociated acid molecules"

   Negation requires special handling, and the scope of negation is often ambiguous. CPL does not currently handle negation, although dealing with negation is relatively well understood in natural language processing.

4. **Vague attributes** ("properties", "characteristic properties") (red)
   "Properties of aqueous solutions of Arrhenius acids are due to H-plus ions"
   "...their characteristic properties disappear altogether"

   The chemistry text sometimes makes "vague" reference to a chemical's properties, without actually stating which specific properties are being referred to. The Halo ontology does not support these kinds of "underspecified" references (e.g., there is no predicate called `characteristic-property`)

5. **coreference** ("react"/"reaction", "hydrogen chloride"/"the chemical") (green)
   "Hydrogen chloride reacts... The reaction produces..."

   Coreference is ubiquitous in text. CPL recognizes simple coreference when the same word is used ("An X...The X"), and also recognizes verbs and their nominalization as coreferential ("... reacts. The reaction..."). In some cases the referring word differs from the introducing word, for example is more general ("a man...the person..."). Resolving these is a little more complex as it requires world knowledge, and the degree of ambiguity can be greater also. Handling this phenomenon is relatively well understood in NLP.

6. **indirect anaphora**: how to resolve references to unmentioned objects. (green)
   "Removing a proton from the acid produces the conjugate base."

   Related to coreference is the phenomenon of indirect anaphora, when a definite reference (e.g., "the base") refers to an object not explicitly mentioned previously, but whose existence is implied. ("John's car wouldn't start. *The engine* was broken."). To identify the referent, and its relation to previously mentioned objects, requires the use of background knowledge. CPL does not support this but this phenomenon has been looked at by (Fan, Barker, and Porter, 2005).

7. how to get **new technical vocabulary** + meanings into the system. (green)
   "NaOH dissociates in water."
   "H2O abstracts the proton from HX"

   Despite its successful use in Vulcan's Halo Pilot project, the Halo KB often was lacking key concepts that were needed (e.g., the meanings of "dissociate" and "abstract" in the example sentences above). Without knowing the meaning of these terms, the formal logic for these sentences is ineffective for question answering. CPL's interpretation algorithm maps words to the "nearest" concept in the ontology, using WordNet to help assign senses to new words. In some cases this results in smart, appropriate word sense disambiguation, but in other cases it can result in seriously over-general choices, or simply fail to find a choice at all. A good interpretation system would be able to recognize when new vocabulary was being used, and be able to back up to read further to identify the meaning of that new vocabulary.

8. how to represent **definitions.** **(green)**
   "Arrhenius acids are defined..."
   "An H-plus ion is a proton with no valence electron."
   (Semi-) formal definitions of concepts are sometimes included in the text. These need to be recognized and translated to specialized representational forms for definitions (e.g., bidirectional implications). This process is relatively well understood. CPL is able to recognized and interpret definitions, but only if stated in an explicit, pre-defined syntactic pattern.

9. how to state that one category is **more general than** another. **(green)**
   "Bronsted-Lowry acids are more general than Arrhenius acids."
   In the simplest case, such statements are simply statements about the type ("isa") hierarchy. CPL will recognize these if a special syntactic pattern is used ("An X is a type of Y"). In other cases, like the above, a phrase like "more general than" can have a highly complex meaning, which is challenging to represent.

10. how to represent **modals/tendencies** like "can". **(yellow/red)**
    "A molecule of a Bronsted-Lowry acid can donate a proton..."
    While statements of potentiality/capability ("can") can be syntactically processed relatively easily, they pose particular representational challenges, as they do not refer to any specific, existing event. The Halo KB does not easily support representation of such statements.

11. how to represent **an argument (proof),** and generalize from it. **(black)**
    "Therefore, the H2O molecule acts as a Bronsted-Lowry base."
    Sometimes the text includes not just statements about the world, but also statements about how those statements can chain together to "prove" something. This knowledge is a kind of meta-knowledge; essentially, the author is conveying a proof tree to the user. Representing proofs in an easily introspectable way is a very challenging problem in AI, independent of any natural language processing issues. In addition, in our work here, this problem is compounded by the "proof" being spread over several sentences, and the "proof" itself being somewhat informal, rather than a strict logical proof.

12. **vagueness** ("is mostly", "nearby", "some") **(red)**
    "An HO3-plus ion sometimes reacts with an H2O molecule."
    "The NH4Cl is mostly solid particles."
    "Some acids are better proton donors than other acids."
    "A weak acid partly transfers the acid's protons to the water."
    "Proton-transfer reactions are governed by the relative strengths of the bases"
    "The solution has a negligible concentration of HCl molecules."
    "An aqueous solution of acetic acid consists mainly of HC2H3O2 molecules"
    "The aqueous solution has relatively few H3O-plus ions"
    Even in Chemistry, vague references are ubiquitous. CPL does not handle vagueness (vague modifiers are ignored), and they present major representational challenges for AI systems.

13. how to compute and represent **differences** **(yellow)**
    "An acid and a base differing only in a proton are called a conjugate pair"

Sometimes the chemistry text makes reference to a comparison between objects, rather than some absolute property. These pose a particular representational challenge.

14. **change over time** (yellow/red)

"The HNO2 molecule becomes the NO2-minus ion."
"The H2O molecule changes into the hydronium ion"
"Acids cause some dyes to change color."

Many knowledge representation systems do not account for changes in time, although the world is intrinsically a dynamic place. For a system to understand change, it needs to support some notion of time-dependent properties and actions, which can change those properties. The Halo KB's underlying knowledge representation language, KM, does support a situation calculus representation of time, and in principle can accommodate such statements, although this aspect is only used in a limited way in CPL so far.

15. How to state and represent **hypothetical** situations. (yellow)

"Assume that H2O is a stronger base than X-minus in Equation 16.9."

In some cases, the text will create a hypothetical scenario and expect the reader to then reason with that scenario. In some cases this can be handled easily, when the hypothetical scenario could just as easily be a real-world scenario. In other cases (not seen here in chemistry, but observed in biology), the hypothetical scenario might be in a world where some basic law of science has changed ("Suppose that DNA was a triple helix, rather than a double helix..."). This again poses major challenges.

16. **algebra:** how to reason with algebraic notions (e.g., formulae) (red)

"OH- is the conjugate base of H2O"

Chemistry is a specialized domain, which involves not just a world of chemicals, but also a world of symbols (chemical formulae and equations). As a result, terms like "OH-" and "H2O" are not just opaque identifiers, but are themselves structured objects to be represented and reasoned about. In this example, the reader is expected to notice (given the context) that the formula H2O is the formula OH- with a proton "added" to it. This kind of reasoning is very challenging to manage. CPL includes machinery for parsing chemical formulae to create structured representations. However, the author has to explicitly indicate when a formula or equation is in the text (by surrounding it in double quotes), and the Halo KB has few accessible primitives for formulae manipulation.

17. **generalized formulae** and equations **(black)**

"In Equation 16.6 the symbol HX denotes an acid."

As an especially complex case of this, our six pages of text sometimes includes generalized formula such as "HX", where "X" denotes some unspecified molecular structure. Recognizing and representing such "abstract chemicals" is extremely challenging.

18. **loosespeak/metonymy** (yellow/red)

"The H2O molecule in Equation 16.5 donates a proton"
"In Equation 16.9 HX dissolves in water."
"Equation 16.9 describes the behavior of a strong acid in water."

Metonymy – using one word to refer to a closely related one – is particularly prevalent in chemistry; or more generally, authors may refer "loosely" (where a literal interpretation of the text

is non-sensical) to objects and events in the domain. In particular, in the domain of chemistry, authors interchange and mix references to molecules, chemicals, and formulae. While a human reader effortlessly untangles these (e.g., above, "Equation 16.5" means "the reaction described by Equation 16.5"), special-purpose machinery is required to recognize and untangle these automatically. CPL does not support metonymy/loosespeak handling, although there has been some work in this area in the literature (Fass 1991, Fan and Porter 2004).

19. **Discourse context (red)**

"Every [Bronsted-Lowry] acid has a conjugate [Bronsted-Lowry] base"

Some NLP systems work only at the sentence level, i.e., treat a paragraph as a "bag of lines". However, often the meaning of one sentence depends on previous sentences in the paragraph. The most obvious examples are pronoun and definite reference resolution, mentioned earlier. Another common example is missing modifiers: A sentence may introduce (say) Bronsted-Lowry acids, then subsequent sentences (such as the one above) simply refer to "acid", implicitly meaning Bronsted-Lowry acids. Finally, the interpretation of a sentence may depend on the overall context in which it is placed (is it an example? A general principle? A motivational sentence). Representation of discourse structure and exploiting it for NLP is essential for much understanding. CPL performs the basic discourse operations (definite reference resolution, consistent word-sense choices across a paragraph), but does not have any representation of the overall discourse structures one might expect in an extended passage of text.

20. Descriptions of **problem-solving methods (red)**

"In any acid-base reaction we can identify two sets of conjugate acid-base pairs."

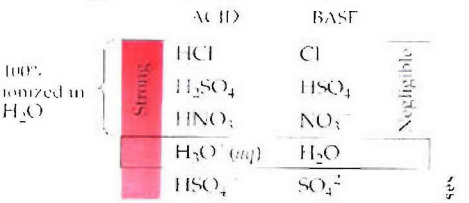
In some parts of the text, the authors describe not events in the real world, but events in the computational world of problem-solving. This problem-solving world has its own vocabulary ("identify", "search", "find", "compare", etc.); the language interpreter needs to identify when this world of problem-solving is being referred to, and construct a problem-solving method from the text description provided. This task is particularly challenging as often problem-solving methods are only partially described in the text, with the reader expected to fill in unstated steps in the algorithm.

21. **Generalization from examples (red/black)**

"In any reaction we can identify two sets of conjugate acid-base pairs. For example, consider the reaction..."

A substantial portion of the six pages of text provides examples of the phenomena being taught. In some cases, they simply exemplify a previously stated general principle (in which case, they contribute little new knowledge). More commonly, though, the general principle is stated in a vague way, or not stated at all, and the reader is expected to refine, or generate, the general principle from the examples. This is a whole research field in its own right.

22. Information in **tables and diagrams (red/black)**



Sometimes key information is simply not stated in text at all (e.g., the table above shows relative strengths of acids and bases). Understanding these requires going beyond NLP to specific technologies for diagram understanding.

4. Analysis II: Core Chemistry Knowledge

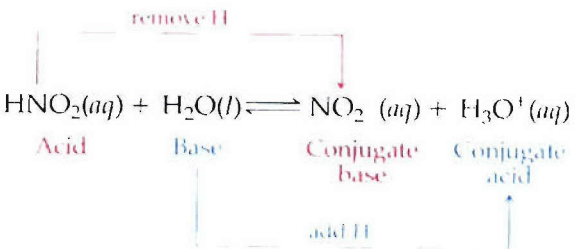
4.1 Introduction

In the first analysis above, we transcribed the 6 pages of text in a relatively literal way into CPL. We have enumerated some of the key problems we encountered in this process. The resulting logic is often imprecise or erroneous for the reasons described, and often relies on concepts, which are not fully represented in the background KB, and do not support extensive inference.

Of particular interest to us is the knowledge required to answer AP-level chemistry questions, the original goal of the Halo KB that we have used. As a result, we now present a second analysis where we look specifically for this knowledge, how it is represented, and how it can be encoded in CPL in an inference-capable way. In this second analysis, the CPL is substantially different from the original text – here we are using CPL as an English-like rule language. This corresponds to taking the large leap (A) to cross the knowledge gap show in Figure 1 earlier. Interestingly, the actual knowledge required for AP questions occupies only a small proportion of the text. In this second analysis, we perform some detective work to find this knowledge in the original text, and look at how it can be expressed in inference-supporting CPL.

4.2 A Quick Chemistry Tutorial

As a preliminary, it is useful to understand a little of the target chemistry knowledge in these six pages which are pertinent to answering AP questions. Essentially, in a reaction between an acid and a base, a proton moves from the acid to the base, as illustrated below:



As a result of this transfer, the original acid becomes a new base (called the acid's "conjugate base"), and the original base becomes a new acid (called the base's "conjugate acid"). Given that the result is a (new) base + acid, a reverse reaction also occurs simultaneously, where the (new) base + acid react to produce the original acid + base. Thus both reactions occur continuously and all 4 substances exist together in the mixture. Equilibrium is established when the rates of each reaction are balanced. If the original acid + base are "stronger" than the new acid + base, then they will react more readily, and the resulting mixture will have a higher concentration of the new base + acid than the original acid + base. We say the equilibrium "lies to the right" (i.e., to the side of the weaker base + acid). Conversely, if the original acid + base are weaker, then they will exist in higher concentration in the mixture than the new base + acid. Here the equilibrium "lies to the left". This is essentially the core knowledge that the 6 pages of text are intended to convey.

One can consider this knowledge to consist of four key methods:

1. Compute the conjugate base (acid) of an acid (base)
2. Identify the strongest base (acid) from a pair of bases (acids)
3. Identify which are acids and which are bases in a given reaction
4. Compute the direction of an equilibrium reaction (left/right)

In addition to this "core" knowledge, there are a lot of "small" facts, e.g., the history of chemistry, examples. A surprising discovery is that the "core" knowledge (at least from the point of view of handling AP questions) is only a tiny proportion (a few sentences) of the overall 6 pages of text.

4.3 Statements of Core Knowledge

One would hope that this "core" knowledge, summarized in the previous Section, is explicitly and clearly presented in the original text, so that we can then transcribe it into CPL (our controlled language). However, this turns out not to be the case. In this Section, we perform some detective work to look for it in the text, and report the results and lessons learned. The bottom line is that the knowledge we want, at least for answering the target AP questions, is only a small fraction of the text, and then rarely stated in the nice explicit form that we would like.

TASK 1: Compute the conjugate base of an acid

Consider task 1, computing the conjugate base of an acid. Essentially, to do this, one removes a proton from the acid, and the result will be the acid's conjugate base. The key sentence in Brown & Lemay, which describes this, is as follows:

(1) "Every acid has a conjugate base, formed by removing a proton from the acid. For example, OH- is the conjugate base of H2O...Similarly, every base has associated with it a conjugate acid, formed by adding a proton to the base."

> absence of a proton are called a **conjugate acid-base pair**.* Every acid has a **conjugate base**, formed by removing a proton from the acid. For example, OH^- is the conjugate base of H_2O, and X^- is the conjugate base of HX. Similarly, every base has associated with it a **conjugate acid**, formed by adding a proton to the base.

Although (1) is (hopefully) sufficient to convey this notion to a person, it is formidable for a computer to process, even if it is rephrased into simplified English. In particular, there are three major challenges here:

1. Mixing the Molecule, Substance, and Algebraic Levels of Description

In Chemistry, there are three different, related worlds, which a scientist reasons with:

- The chemical (macro-level) world of substances, test-tubes, mixtures, etc.
- The atomic world of molecules, electrons, atoms, etc.
- The algebraic world of equations, formulae, terms, coefficients, subscripts, etc. This algebraic world includes spatial references e.g., the left-hand side of a reaction.

These worlds are frequently mixed together in text in a way, which does not make sense if taken literally. Although people effortlessly disentangle these worlds, in fact so effortlessly that the mixing goes unnoticed, they pose formidable problems for the computer. For example, in (1) "formed by

removing a proton from the acid" mixes the atomic ("proton") and chemical ("acid") levels together. Similarly "OH- is the conjugate base of H2O" introduces the algebraic world.

This mixing is ubiquitous in chemistry, not an idiosyncrasy of this particular sentence, and it causes a major challenge for language interpretation. As we describe in more detail later, rules the hand-built Halo Pilot KB untangle this mixing so linguistic shorthand ("the acid on the left") is expanded into its precise form ("the acid denoted by the formula on the left side of the equation denoting the reaction"). Untangling this mixing alone accounts for a large (around half) of the complexity of the Halo KB.

2. Understanding Algebraic Manipulation

A substantial part of chemistry involves algebraic manipulation of formulae. Handling formulae in general poses significant challenges for language understanding, because a formula is not just an opaque token denoting some object in the real world - rather, it is itself an object (in the abstract world of symbol systems), with appearance, shape, parts, orientation, etc. Thus, the formula itself needs to be parsed - in a way defined by the rules of the formula's algebra - to create a representation of that formula. Just as AI systems reason about objects in the real world by creating representations of those objects, relationships, and actions that can manipulate them, so AI systems can only reason about formula if they can create representations of those formulae, relationships, and actions that can manipulate them. Above, when the authors wrote

"OH- is the conjugate base of H2O"

they expect the reader to see that the formula OH- is the algebraic result of the "removing a proton" operator applied to the formula H2O.

3. Describing Declarative Knowledge Procedurally

A conjugate base is "formed by removing a proton from the acid." This phrase is a procedural description of what is essentially a declarative constraint ("conjugate base = acid - proton"). The reader needs to understand the declarative definition, but for a computer to derive this automatically from the procedural statement is challenging. This phenomenon arises in several other places in the text also.

Another way of viewing this is that the text is describing a (simple) problem-solving method (PSM), i.e., describing a procedure by which the user can work out the conjugate base of an acid. The PSM explains how the (unstated) declarative knowledge is made operational, so that it can be used to actually solve a problem, but for a more general understanding it is useful to know that declarative knowledge. This also happens in the domain of physics: for example, some text can explain that one can find the force F on an object by multiplying its mass m times its acceleration a; but a more general solution would be to teach that F = m x a, plus teach general rules about solving equations to find an unknown value from known values.

TASK 2: Identify the strongest base from a pair of bases

In this particular text, the required information is present in a table, making it inaccessible to language processing. Our CPL encoding simply enumerated entries in this table.

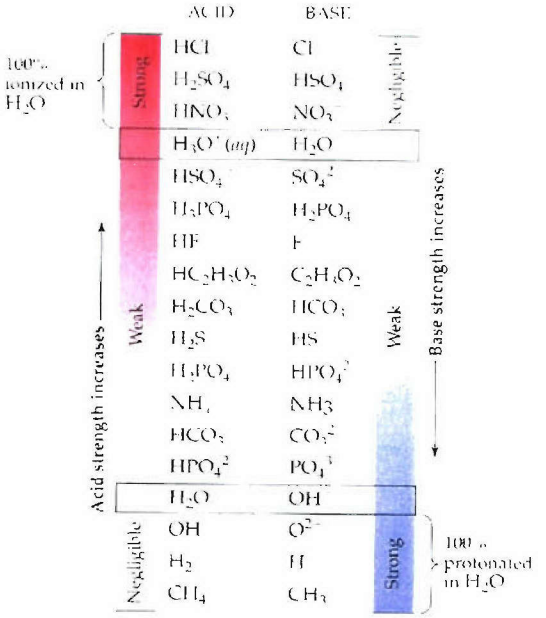


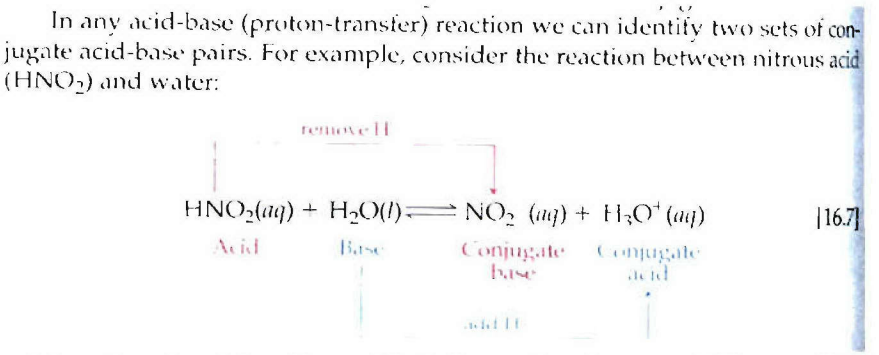
Figure 2: Relative strengths of acids and bases are conveyed pictorially in the text.

TASK 3: Identify which are acids and which are bases in a given reaction

This task requires looking at the chemicals on the left- and right-hand side of a reaction, and finding pairs where a right-hand chemical is the conjugate base of a left-hand chemical (and vice versa). In this text, this is described largely by an example diagram. The authors write:

> "In any acid-base (proton transfer) reaction we can identify two sets of conjugate acid-base pairs."

And then present a diagram from which the user is meant to "see" how to do this. Again there is no text, which explicitly states the algorithm.



Automatically acquiring the algorithm from this information is very different. Rather, the reader is expected to already know basic notions such as "trying different pairings of left- and right-hand chemicals", and recognize that this is what is being asked for here. There is a basic, iterative problem-solving method being used here, which is not stated in the text, but rather is assumed to be already known. For a computer to understand this text, it similarly needs to already know and recognize the application of this algorithm in this context.

TASK 4: Compute the direction of an equilibrium reaction (left/right)
Again the text is indirect about how to do this. It provides two examples, and then says:

> "From these examples, we conclude that in every acid-base reaction the position of the equilibrium favors transfer of the proton to the stronger base"

Never is the algorithm explicitly specified; rather, the reader is meant to induce it from the examples. In addition, this sentence provides a fine example of how the chemical ("base"), atomic ("proton"), and algebraic ("position") worlds are casually mixed together in the same sentence, as illustrated below:

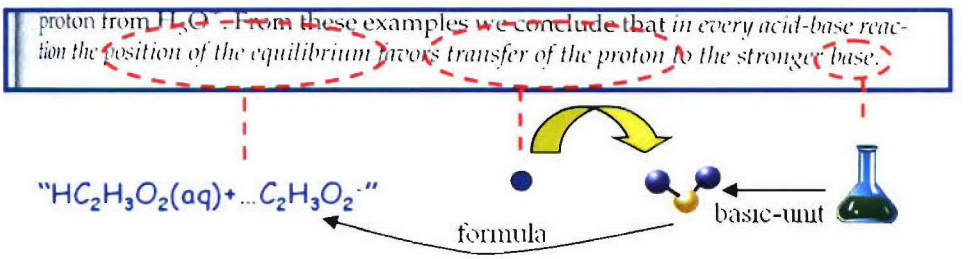


Figure 3: This sentence illustrates how the author has mixed the algebraic ("position"), atomic ("proton") and chemical ("base") perspectives. Although people can effortlessly disentangle these, they still remain challenging for a machine.

4.4 Encoding in CPL

Because of the paucity of clear, precise, declarative text, our CPL encodings of this knowledge is substantially different to the text. This particular version of the CPL is inference-supporting, but it is not plausibly derivable from the original pages through automated means. This is path (A) across the "gap" described earlier, and essentially treats CPL as an English-like rule language:

TASK 1: Compute the conjugate base of an acid

As we describe later, the background KB does not have the basic primitives for formula manipulation that we need. As a result, we encoded the conjugate acid/base computations as a simple lookup procedure:

```
The conjugate acid of Cl-Minus is HCl.
The conjugate acid of the water is H3O-Plus.
<and so on>
```

TASK 2: Identify the strongest base from a pair of bases

As described earlier, this information was presented in a table in the chemistry text, and so was transcribed into rules such as:

```
Water is a stronger than Cl-Minus.
<and so on>
```

where "stronger than" is interpreted as meaning stronger-base-than. The transitivity of "stronger than" is assumed in the text, but needs to be spelt out in a KB. This is a good example of assumed commonsense which the KB is missing:

```
IF a first chemical entity is stronger than a second chemical entity
AND the second chemical entity is stronger than a third chemical entity
THEN the first chemical entity is stronger than the third chemical entity.
```

TASK 3: Identify which are acids and which are bases in a given reaction

As discussed earlier, the full expansion of this knowledge into CPL is complex, in particular being precise about the chemical/atomic/algebraic worlds and making the mappings between them explicit:

```
IF there is an equation of a reaction
AND a first chemical entity has a chemical formula
AND a second chemical entity has a second chemical formula
AND the first chemical formula is part of the left side of the equation
AND the second chemical formula is part of the right side of the equation
AND the first chemical entity is the conjugate base of the second chemical
entity
THEN the first chemical entity is playing a base role
AND the second chemical entity is playing an acid role.
```

The CPL implicitly executes a "generate and test" algorithm, trying different pairs of chemicals on the left- and right-hand side of the equations until a solution is found, by the non-deterministic clauses "the first (second) chemical formula is part of the left (right) side of the equation". As there are multiple (two) left-hand formulae, and two right-hand formulae, the inference engine will backtrack until a solution is found which satisfies the last clause "the first chemical entity is the conjugate base of the second chemical entity". A second, similar CPL rule provides the inverse of this rule, where the relation is conjugate acid.

TASK 4: Compute the direction of an equilibrium reaction (left/right)

Finally, the CPL rule for computing the equilibrium side, which uses the output of the previous rules, looks as follows:

```
IF there is an equation of a reaction
AND a first chemical entity has a chemical formula
AND a second chemical entity has a second chemical formula
AND the first chemical formula is part of the left side of the equation
AND the second chemical formula is part of the right side of the equation
AND the first chemical entity is playing a base role
AND the second chemical entity is playing a base role
AND the first chemical entity is stronger than the second chemical entity
THEN the direction of the reaction is right
AND the equilibrium side of the reaction is right.
```

Again, a second, similar CPL rule exists for the opposite case, when the reaction is to the left.

4.5 Summary

The CPL rules above support inferencing and chain together to answer AP questions about acid-base reactions. However, as we have described, they are substantially distant from the original text, which for the key sentences hits several of the challenges enumerated in the previous Section. The "knowledge gap" is thus still a problem, and these different formulations of CPL in Section 3 and

Section 4 illustrate the size of this gap. We provide an overall assessment of the gap later in this document.

5. Knowledge Integration and Extensible Knowledge Bases

5.1 Introduction

Having discussed at length the chemistry text, and the resulting linguistic and semantic issues, we now turn our attention to the background chemistry KB (the Halo KB) that we are using. The KB was handcrafted as part of Vulcan's Halo Pilot with the explicit goal of answering chemistry AP questions, including questions about acid-base equilibria, the topic of our six pages. In the formal Halo evaluation, the KB was found to perform well (Friedland et al, 2004).

Our goal with our CPL-generated knowledge is that it is integratable with the existing chemistry KB. As part of our experimental methodology, our goal was to surgically remove the hand-written acid-base knowledge from the original Halo KB, add in the CPL generated knowledge, and compare performance.

Ideally, our CPL generated knowledge would look similar to the equivalent hand-built knowledge in the KB, so we can just remove the latter and insert the former. Unfortunately, this is not the case; the hand-built knowledge is highly complex and intertwined. In this section we study the previous encoding of this chemistry knowledge, look at why it is complex and why this makes knowledge integration hard, and then reflect on how we would like that original knowledge to have looked so that knowledge integration would be easier. It is important to note that the original KB was never intended to support knowledge integration, so our goal is not to be critical of the KB. Rather, we are trying to do something with that KB for which it was never intended, and we describe the problems encountered. The bottom line is that, like software, a knowledge base needs to be designed for reuse/extension. We identify some principles for doing this in Section 5.4.

5.2 Case Study 1: Conjugate Acid-Base Calculations

Task 1 in the earlier analysis is to compute the conjugate base of an acid, describe in the text as "removing a proton" from the acid. For example, the conjugate base of H3O+ is H2O, as H2O denotes a molecule with one less proton than H3O+.

How does the Halo KB perform this computation? Ideally, it would have some representation of the algebraic operation S of "subtracting a proton", which could then be applied to a chemical formula to yield a new one. (Or more generally, subtracting X). If this were the case, then CPL could plausibly map the word "removing" in the text's "removing a proton" to the operation S, and then possibly be able to perform this computation.

In fact, this subtraction operation is not a primitive in the Halo KB, rather the Halo KB contains a highly complex, hand-written method for computing conjugate acids, expressed in the language KM:

```
1 (every Compute-Conjugate-Acid has
2    (input ((a Chemical with (plays ((a Base-Role))))))
3    (parent_formula ((the term of
```

```
 4                              (the nested-atomic-chemical-formula of
 5                                  (the has-basic-structural-unit of
 6                                       (the input of Self))))))
 7    (target-unit

 8          ((if (the parent_formula of Self) then
 9            (:set
10             (#'(LAMBDA ()  (GET-CONJUGATE-ACID-ATOMIC-FORMULA-BACK
11                  (KM0 '(|the| |parent_formula| |of| |Self|))))))))))
12    (output
13     ((if (oneof (the input of Self) where (It isa H2O-Substance))
14       then
15       (a H3O-Plus-Substance)
16       else
17          ((forall
18            (allof2 (the target-unit of Self)
19                    where
20                    ((not (It2 = (the parent_formula of Self)))))
21            (the output of
22             (a Identify-Chemical with
23              (input
24               ((a Chemical with
25                 (has-basic-structural-unit
26                  ((the output of
27                   (a Identify-Chemical-Entity with
28                    (input
29                     ((a Chemical-Entity with
30                       (nested-atomic-chemical-formula
31                        ((a Chemical-Formula with
32                          (term (It))))))))))))))))))))
```

Despite the opacity of this code, it is worth explaining what is going on here. In this frame representation language, Compute-Conjugate-Acid (line 1) is a frame, with slots input, parent_formula, etc., that can be queried. Querying a slot causes the expression on that slot to be evaluated to find the answer. in this case, this frame takes as input a chemical (line 2) and produces as output a new chemical (line 12), which is the conjugate acid of the input chemical. The actual computation of conjugate acid is performed by an external Lisp procedure, called in lines 10-11, which takes as input a formula and outputs the conjugate acid's formula (i.e., implements "adding one proton"). The bulk of the representation here, then, is converting between chemicals, molecules, and formula. The original input chemical (line 2) is converted to its basic molecule (line 5), to its formula (line 4), then passed to the Lisp procedures (line 11). The resulting formula is then coverted back to a molecule (line 29), and then to a chemical (line 24). The references to Identify-Chemical (line 22) and Identify-Chemical-Entity (line 27) ensure that the new chemical and new molecule have been properly classified in the knowledge base. This sequence of transformations is shown schematically as follows:

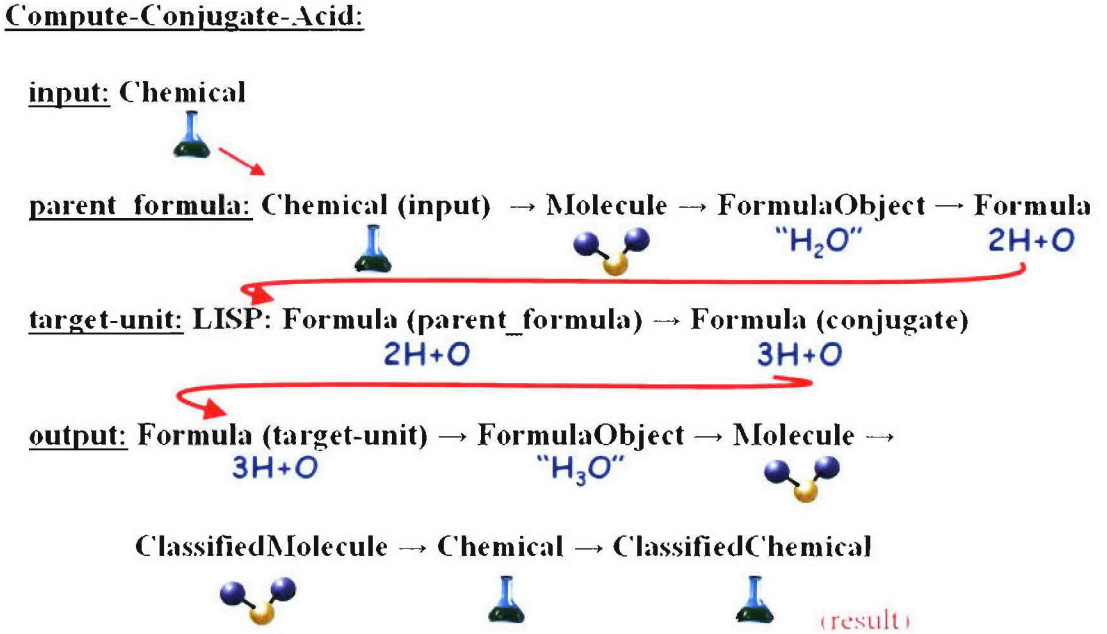


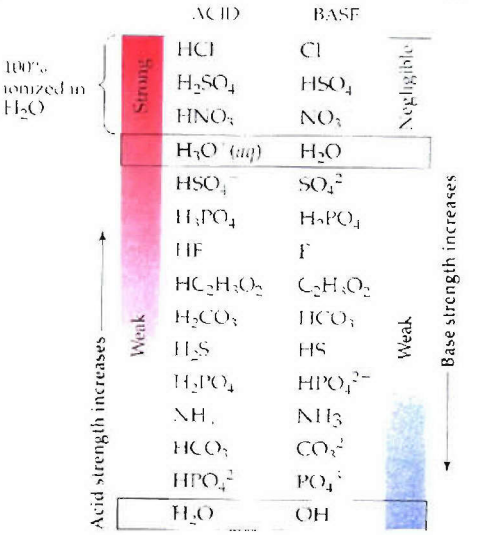
Figure 4: Much of the logic in the Halo KB is for untangling the chemical/ molecule/formula distinction. This figure illustrates the conversions involved in the Compute-Conjugate-Acid method, shown earlier.

There are several important points to draw from this:

1. Clearly such syntactic complexity is knowledge representation is a major obstacle for modification and integration of new knowledge. What would be preferable would be **syntactically simpler structures** that are more amenable to automated manipulation. It is highly unlikely that a NLP interpretation system will be able to generate structures of similar complexity automatically. Rather, some alternative means of representing and/or factoring this knowledge are needed.
2. The representation here is of a special-purpose method for conjugate acid calculation, while what we would have preferred would have been to see representations of general algebraic manipulation operators (e.g. "add a proton"), and then conjugate acid calculation defined in terms of those; in other words, **represent reasoning primitives, and then represent specific operations using those primitives**, rather than combining the two into a single procedure.
3. The representation is procedural in nature, saying how to compute the conjugate acid without declaratively specifying what the conjugate acid is. What would be preferable would be to **separate the declarative knowledge** ("Acid = base + proton") **from general methods for reasoning with that knowledge** (here, algebraic equation solving).
4. The bulk of the KM code above deals with what can be considered the metonymy problem in chemistry: Converting between chemicals, molecules, and formulae. Although people can effortlessly identify what is intended in text, the logical formulation must painstakingly spell out the conversions from one world to another in order to function. This phenomenon is not specific to this particular procedure; from an informal inspection of the Halo KB, approximately half of the content in the KB is dealing with this one specific problem. Clearly **methods for dealing with metonymy**, at least in chemistry, are critical for automated knowledge acquisition.

5.3 Case Study 2: Comparing Acid Strengths

Recall that relative acid strengths are encoded in a table in the original text:



In the Halo KB, rather than represent a partial ordering on strength, the KB authors decided to use three absolute, qualitative, strength values (called the acid's "intensity"): `*strong`, `*weak`, and `*neglible`. The table itself is encoded using a large nested if-then structure:

```
 1 (every Acid-Role has
 2  (intensity (
 3    (a Intensity-Value with
 4     (value (
 5      (:pair
 6       ;; Case statement for Acids.
 7       (if ((the played-by of Self) isa Ionic-Compound-Substance)
 8         then
 9        (if (((the played-by of Self) isa HCl-Substance) or
10          ((the played-by of Self) isa HBr-Substance)    or
11          ((the played-by of Self) isa HI-Substance)     or
12          ((the played-by of Self) isa HClO3-Substance)  or
13          ((the played-by of Self) isa HClO4-Substance)  or
14          ((the played-by of Self) isa H2SO4-Substance)  or
15          ((the played-by of Self) isa HNO3-Substance))
16          then
17          *strong
18          else
19          (if (((the played-by of Self) isa H3PO4-Substance) or
20            ((the played-by of Self) isa HF-Substance)       or
21            ((the played-by of Self) isa HC2H3O2-Substance)  or
22            ((the played-by of Self) isa H2CO3-Substance)    or
23      ....
```

Lines 9-15 enumerate strong acids ("`if ... then *strong`", line 17). Lines 19 and onwards enumerate weak acids, and so on. To decide which acid is strongest, the following procedure is used:

```
 1 (every Compare-Relative-Strengths-of-Acids has
 2  (output ((if (((the1 of (the value of (the intensity of
 3                    (the Acid-Role plays of
 4                      (the first of (the input of Self))))))
 5                  = *strong)
 6                 and
 7                 ((the1 of (the value of (the intensity of
 8                    (the Acid-Role plays of
 9                      (the second of (the input of Self))))))
10                  /= *strong))
11                then
12                (the first of (the input of Self)))
13           (if ((....
```

This procedure essentially encodes a lookup table for a three-valued qualitative scale:

- **IF** X is strong & Y is not strong **THEN** X is strongest (lines 2-12)
- **IF** X is not strong & Y is strong **THEN** Y is strongest
- **IF** X is weak & Y is negligible **THEN** X is strongest
- **IF** X is negligible & Y is weak **THEN** Y is strongest

Lines 2-12 above encode the first rule; subsequent lines (not shown) encode the remaining three rules.

Again, there are some important points to draw from this:

1. The nested if-then structure encoding the qualitative acid strengths is not easy to automatically extend, due to its syntactic complexity. A more preferable encoding would be a larger number of ground assertions.
2. The `Comparing-Relative-Strengths-of-Acids` computation combines general knowledge about reasoning with ordered scales with specific knowledge about acids. What would be preferable would be to separately represent knowledge about reasoning with ordered scales, and then apply that knowledge in the specific context of reasoning about acid strengths. In addition, the general knowledge about ordered scales would be generalized to N-valued rather than 3-valued scales.

5.4 Five Principles for an Extensible Knowledge Base

It is important to remember that the Halo KB was not built with extensibility in mind. However, our look at its structure has revealed several challenges that it presents if it were to be automatically extended. Based on this analysis, we identify five principles for extensibility as follows:

5.4.1 Handle Loosespeak/Metonymy

As described, a major obstacle in chemistry is handling loosespeak/metonymy, namely where a "literal" or "direct" interpretation of the language does not make sense. In particular the chemical/molecule/formula distinction is a major source of loosespeak/metonymy in chemistry, e.g., "the acid on the left" means "the acid denoted by the formula on the left side of the equation of the reaction). A substantial part of the Halo KB is devoted to untangling this. The precision that logic requires of our written representations is a fundamental barrier to robustness.

Work by Fass (Fass, 1991), Fan and Porter (Fan and Porter 2004), and others have demonstrated that, with suitable background knowledge, metonymy in language can be handled automatically.

Traditionally, this has been handled at language interpretation time, where metonymous text is expanded into precise logic. An alternative, which we also mention here, is to preserve metonymy in the KB itself, and then have it resolved at reasoning time. For example, following this approach, the Compare-Relative-Strengths-of-Acids method shown earlier could be rewritten in a simpler form such as:

```
-----------------------------------------------------------------
(every Compare-Relative-Strengths-of-Acids has
  (output (
   (if   ((the intensity of (the first of (the Chemicals))=*strong)
    and  ((the intensity of (the second of (the Chemicals))/=*strong)
    then (the strongest of (the Chemicals))
                 = (the first of (the Chemicals)))))
-----------------------------------------------------------------
```

where it then left to the reasoning engine to realize that "the intensity of the chemical" is a shorthand for a more complex expression, and expand it accordingly. The advantage of this is that the linguistic and logical structures become closer, making generation of the latter from the former more plausible. The disadvantage is that the formal structures are now only "semi formal", with some interpretation deferred to the reasoner, making proving formal properties of the representation difficult. A third alternative would be to retain both the metonymous and fully expanded representations, keeping them synchronized, the former being used for integrating new knowledge into, and the latter being used for reasoning.

5.4.2 Cleanly Separate Declarative and Procedural Knowledge

As describe earlier, the Halo KB contains procedures for solving specific problems:

- Computing a conjugate acid from a base
- Identifying the strongest acid from a pair of acids

In fact, these procedures can be viewed as a specific application of general methods (chemical formula manipulation, reasoning about partial orders) to specific data. A preferable encoding would be to encode the general methods separately from the data that they are being applied to, and then implement the specific procedures using these general methods. These are good examples of where explicit design for reuse would help.

This lesson repeats significant lessons and methods developed in the wake of the '80s boom in Expert Systems. In particular, Clancey showed how many expert systems could be viewed as implicitly applying general problem-solving methods (PSMs) to specific data, and illustrated how one such system, Mycin, could be recast in this way so that the general PSM was explicit, and hence reusable for other tasks (Clancey, 1984; Clancey, 1992). This work was a catalyst for KADS, an entire methodology for designing and constructing expert systems in reusable ways (Wielina et al, 1992), and an entire subfield of AI devoted to identifying and creating libraries of reusable problem-solving methods, e.g., (Chandrasekaren 1986)

5.4.3 Create Elaboration Tolerant Representations

In (McCarthy, 1998), McCarthy identified the notion of "elaboration tolerance" for knowledge representations. A representation is elaboration tolerant to the extent that it is convenient to syntactically modify a formalism to take into account new phenomena or changed circumstances. In the simplest case, a representation has high elaboration tolerance if it can be semantically extended by

simply adding new axioms to it. The lesson from this, and from our earlier examples, is that syntactic organization matters. The earlier example of an acid's strength being encoded as a giant if-then statement is a good example of an elaboration *in*tolerant representation, as complex syntactic manipulation is required to (for example) add a new strong acid to the representation. Better would be to encode this knowledge as a set of separate clauses, for example:

```
-----------------------------------------------------------------
intensity(HCl-Substance, *strong)
intensity(HBr-Substance, *strong)
intensity(HI-Substance, *strong)
intensity(HClO3-Substance, *strong)
intensity(HClO4-Substance, *strong)
intensity(H2SO4-Substance, *strong)
intensity(HNO3-Substance, *strong)
...
intensity(HF-Substance, *weak)
intensity(HC2H3O2-Substance, *weak)
intensity(H2CO3-Substance, *weak)
...
-----------------------------------------------------------------
```

If this were done, then a new strong acid could be added to the KB simply by adding a new clause, rather than attempt to have the computer surgically alter a giant if-then rule. More generally, devising structures where semantic changes can be instigated by simple syntactic changes is an important goal in creating extensible representations.

5.4.4 Use a Linguistically Motivated Ontology

A key challenge in NLP is word sense disambiguation (WSD), mapping from the English words/phrases to knowledge-base concepts. In our work here, many words map straightforwardly to concepts (e.g. the word "HCl" maps to the concept HCl-Substance). However, in some cases key concepts were missing in the KB, or the KB presented a different conceptual view of the world to that used in language. As an example of the former, although the Halo KB contains a notion of acid strength, and a method for computing the strongest of two acids, it does not include a "stronger acid than" relationship, and as a result this phrase has no obvious translation into the Halo KB ontology. As an example of a different conceptual view, the Halo KB attaches the notion of equilibrium direction (left/right) to the concept of a reaction, while the text describes equilibrium direction in terms of an equation. This mis-match complicates translation from the English text to knowledge base concepts. The general lesson is that the more that the KB is in line with linguistic notions, the fewer such complications will arise.

5.4.5 Develop Error-Tolerant Reasoning Methods

Finally we highlight the challenge of reasoning with a knowledge base that inevitably contains errors, approximations, and imprecision's. Most of the work on formal reasoning in AI has assumed a correct knowledge base, and most of the work on knowledge base construction has tried to eliminate errors through sheer hard work. However, for KBs beyond a certain size, and for KBs partially constructed using automated methods, errors are inevitable, and need to be seriously addressed. Systems need to be able to both introspect on their knowledge to identify and reduce errors, and also perform inference in new ways to better tolerate errors (e.g., rather than backward chaining to find a single proof path, perform additional tangential reasoning to ensure that the intermediate and final conclusions are

plausible with respect to other facts and constraints that are known). One can view this as a kind of "crystallization" process of creating a model of the world most consistent with data and background knowledge, rather than myopic search for a single chain of rules from facts to a possible conclusion.

6. Evaluation and Quantification of the Gap

6.1 Performance on AP Questions

Our original goal was to compare the performances of the CPL-generated knowledge with that of the original Halo KB on AP Chemistry questions. In fact, because of the relatively short amount of text we are dealing with, we can predict the performance on any given question by analysis. As described in Section 3, there are essentially 4 key methods, which are required for solving AP questions targeted at this text:

1. Compute the conjugate base (acid) of an acid (base)
2. Identify the strongest base (acid) from a pair of bases (acids)
3. Identify which are acids and which are bases in a given reaction
4. Compute the direction of an equilibrium reaction (left/right)

By inspection of the CPL-generated logic (Section 3), and the Halo KB (Section 4), we can directly compare the generality of the encoded solutions, as summarized below:

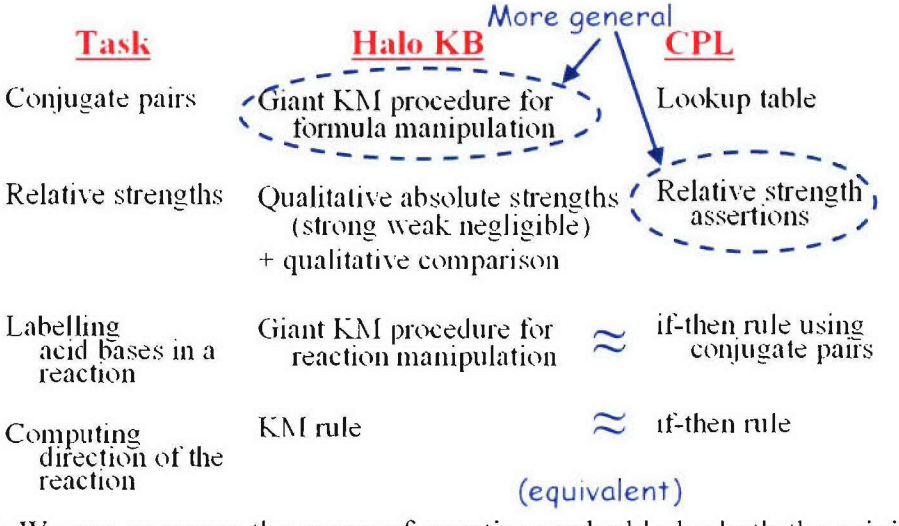


Figure 5: We can compare the scope of questions solvable by both the original (Halo KB) and CPL encodings of the key chemistry knowledge, by analyzing the generality of the encoded knowledge.

For the first task, of computing conjugate pairs, the CPL encoding uses a lookup table, which is thus limited to just those chemicals in that table. The Halo KB, however, uses a complex procedure for formula manipulation, which can be applied to any acid/base. Thus (assuming no software bugs) we can predict that in some cases the Halo KB will be able to answer questions, which the CPL encoding cannot, for this particular task.

Conversely, for the second task, the Halo KB encodes relative acid strengths using just three qualitative values, while the CPL encoding uses a fully ordered scale of relative strengths. We can thus predict that in some cases the CPL encoding will be able to recognize relative strengths, which the Halo KB will not (in cases were the two acid/bases have the same qualitative strength in the Halo KB's three-valued intensity scale).

Finally for tasks 3 and 4, labeling acids/bases in a reaction and computing the reaction direction, we can see by inspection that the two formulations in Halo and CPL implement essentially the same procedure (albeit with very different appearances). We can thus conclude that the performance of these procedures will be the same, again ignoring software bugs.

The net result of this analysis is that, for any given sample of AP questions, we can identify by analysis, which KB will score highest. While we could in principle perform this experiment for real, this would tell us little beyond the statistical distribution of question types in the AP exam, something that is tangential to our concerns here. Rather, the most significant thing we have learned from our work is the size and nature of the "knowledge gap" to be bridged. As a result, we have instead focused our concluding analysis on a rough quantification of this gap, presented in the next Section.

6.2 A Quantification of the Knowledge Gap

In Section 3.4 we listed a number of phenomena contributing to the knowledge gap. In this Section, we aggregate these into some broader categories, and present a rough quantification of their prevalence in the 6 pages of chemistry text that we have studied. For comparison, we also compare this with their prevalence in six pages of grade-school level biology text about a different subject (the structure and function of the heart), to give some indication of which phenomena are domain general and which are accentuated in college-level chemistry.

The 22 categories of Section 3.4 were aggregated as follows:

- Idioms/special-purpose phrases (item 1 earlier)
- Generics (item 2)
- Knowledge representation challenges (items 3-15)
- Algebra/mathematics (items 16-17)
- Loosespeak/metonymy (item 18)
- Discourse context (item 19)
- Problem-solving method descriptions (item 20)
- Learning from examples (item 21)
- Tables and diagrams (item 22)

The prevalence of these phenomena was identified by counting the numbers of sentences in which they occur. It is important to note that this is a very loose quantification given the relatively small amount of text (six pages) looked at in each science.

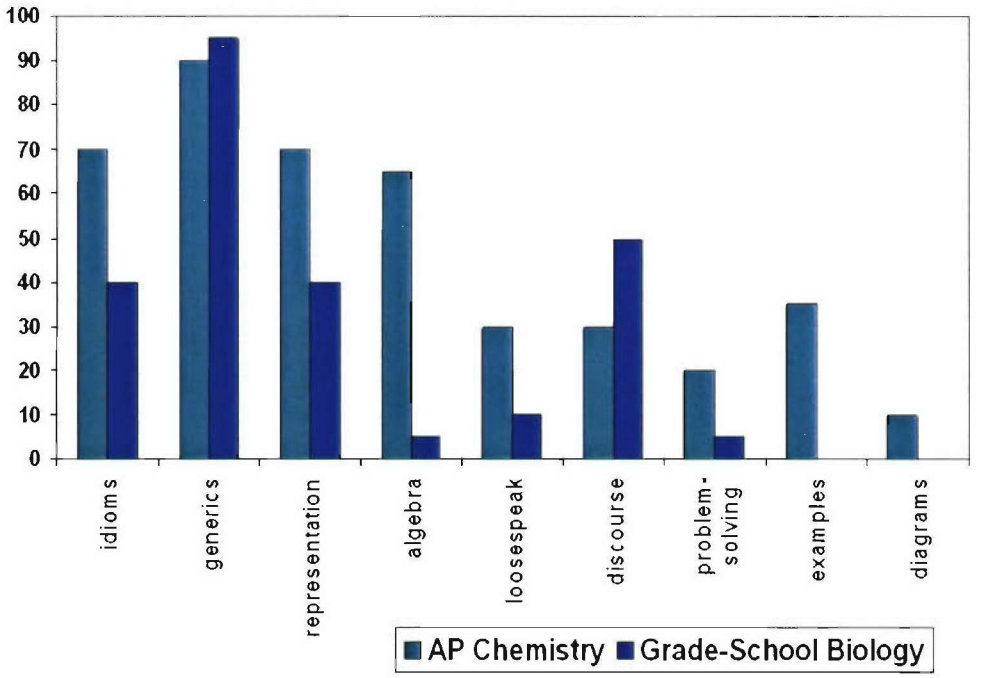


Table 1: Quantification of the relative frequency (% of sentences) of the nine major challenges, as seen in our target chemistry text and a comparably sized biology text.

There are some interesting, albeit tentative, conclusions, which can be drawn from this data. First, AP level chemistry text is considerably harder to interpret computationally than grade-school level biology text. Part of this stems from the educational level of the text: grammatically, the biology text contained shorter, simpler sentence structures, which were more likely to stand on their own than the chemistry text. Use of idiomatic phrases/phrases with idiosyncratic meaning was less common in the biology text. Semantically too, the complexity of the knowledge being communicated is higher at the AP level, resulting in a higher number of "representational challenges" to address than for grade-school level text. As a second dimension of comparison, the discipline makes a difference also: Biology is a subject concerned with structure, function, and their relationship in real-world settings - these are things which AI systems can model relatively well. In contrast, chemistry, at least at the AP level, includes reasoning at the molecular as well as real-world level, and the use of equations and formulae is central to modeling what is happening. All these pose extra challenges for machine understanding of text. Thus, it appears that some of the complexities of chemistry are not universal, but rather peculiar to that particular science (or class of sciences), and that machine understanding may prove less challenging in other domains.

7. Summary and Conclusions

Reading is a primary way in which human beings learn, and ultimately will be a primary way that machines will learn also. In this project, we have conducted a detailed investigation into the opportunities and challenges of having a machine read to learn, focusing on part of the domain of chemistry, and simplifying the challenge by working with controlled, as opposed to full, English language. Our work shows that while grammatical simplification of the language helps somewhat in language understanding, significant challenges still remain. As described in Section 6.2, and detailed in Section 3.4, we identified **9 major challenges**, both linguistic and semantic, that were encountered in

this domain, and still remain despite extensions that we have made to our CPL interpreter during this project:

- Idioms/special-purpose phrases
- Generics
- Knowledge representation challenges
- Algebra/mathematics
- Loosespeak/metonymy
- Discourse context
- Problem-solving method descriptions
- Learning from examples
- Tables and diagrams

Full descriptions and examples of these were given earlier in Section 3.4. In addition, we have studied the challenge of integrating the CPL-generated logic into a pre-existing KB, and discovered several obstacles to doing so with a knowledge base that was not designed with extensibility in mind. As a result, we have identified five recommendations for **constructing extensible knowledge bases** as follows:

- Handle Loosespeak/Metonymy
- Separate Declarative and Procedural Knowledge
- Create Elaboration Tolerant Representations
- Use a Linguistically Motivated Ontology
- Develop Error-Tolerant Reasoning Methods

Again, full descriptions and discussions of these were presented earlier in Section 5.4.

Despite the variety of these issues, they essentially all require the use of extensive, prior knowledge - both linguistic and semantic - to help guide and correct the interpretation. This creates a kind of "catch 22": reading can add knowledge, but only if there is reliable knowledge there to begin with to guide that reading. If a system had sufficient prior knowledge about plausible relationships between objects in a domain, for instance, it would then be able to recognize and correct metonymy or other imprecision's in the input language. How can a system acquire such knowledge in the first place? While some hand-coding may be feasible, it seems clear that only practical way forward is through some bootstrapped/looping approach, in which some initial knowledge supports at least some reading, which then augments knowledge, supporting further reading etc. In terms of our original metaphor of the knowledge gap, rather than thinking of reading as a task of "climbing a cliff", it is perhaps better thought of as a two way process, where knowledge flows down from the top, providing expectations, context, prior knowledge, and hypotheses for interpreting language, and new information from language provides fragments of new knowledge, examples, confirmations, and refinements to existing knowledge:

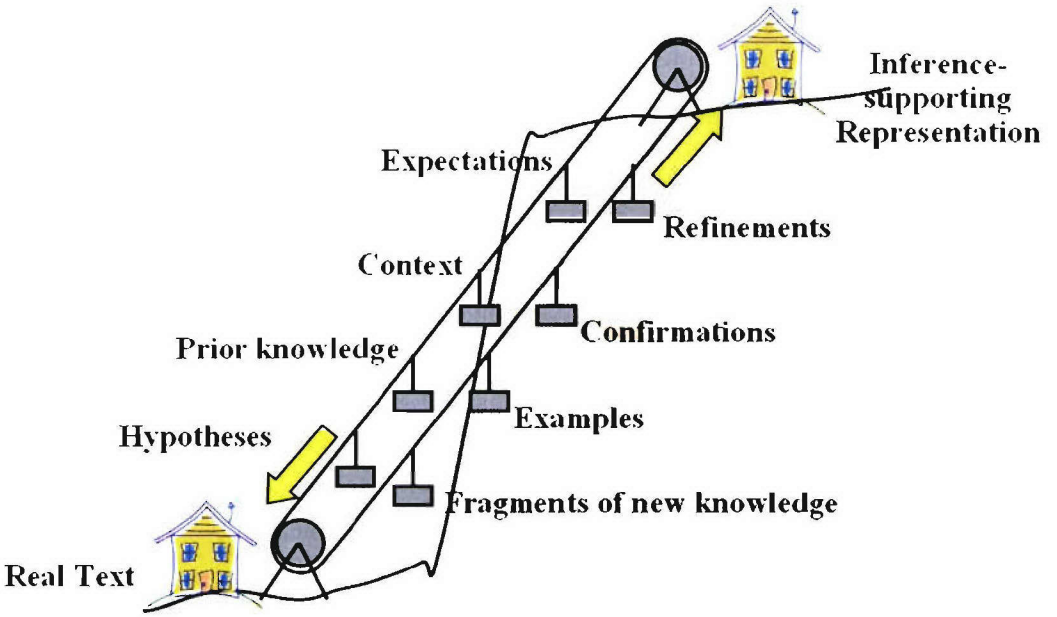


Figure 6: Boostrapping/looping as a means of bridging the gap, whereby knowledge guides reading and reading supplies new knowledge.

Thus rather than thinking of language processing as a pipeline, viewing it in terms of a symbiotic relationship between language and knowledge is perhaps a more appropriate road to success, and would provide a powerful basis for addressing the challenges involved. We are excited by this possibility, and are optimistic that technologies for reading to learn will continue to develop rapidly in the future.

References

K. Barker, V. Chaudhri, S. Chaw, P. Clark, J. Fan, D. Israel, S. Mishra, B. Porter, P. Romero, D. Tecuci, P. Yeh. A Question-Answering System for AP Chemistry: Assessing KR&R Technologies. In *Proc 9th International Conf on Knowledge Representation and Reasoning (KR'04)*, 2004, AAAI Press.

T. Brown, E. LeMay, H. Bursten. Chemistry: The Central Science. NJ:Prentice-Hall, 2003.

B. Chandrasekaren. Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design. In IEEE Expert, pp 23-30, 1986.

W. Clancey. Classification Problem Solving. In AAAI'84, pp 49-55, 1984.

W. Clancey. Model Construction Operators. In Artificial Intelligence 53 (1), pp1-116, 1992.

P. Clark, P. Harrison, T. Jenkins, J. Thompson, R. Wojcik. Acquiring and Using World Knowledge using a Restricted Subset of English. In: *The 18th International FLAIRS Conference (FLAIRS'05)*, 2005.

J. Fan, K. Barker, B. Porter. Indirect Anaphora Resolution as Semantic Search. *Proc. 3rd International Conference on Knowledge Capture*, pp153-160, 2005.

D. Fass Met*: A Method for Discriminating Metonymy and Metaphor by Computer, in *Computational Linguistics* 17 (1), 49-90, 1991.

N. Friedland, P. Allen, M. Witbrock, G. Matthews, N. Salay, P. Miraglia, J. Angele, S. Staab, D. Israel, V. Chaudhri, B. Porter, K. Barker, P. Clark. Towards a Quantitative, Platform-Independent Analysis of Knowledge Systems. In *Proc 9th International Conf on Knowledge Representation and Reasoning (KR'04)*, 2004, AAAI Press.

J. McCarthy. Elaboration Tolerance. In Proc 4th Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense '98), 1998.J. Fan, B. Porter. Interpreting Loosely Encoded Questions. In *AAAI*, 2004.

J. Thompson. The CPL User Guide. Boeing Phantom Works Technical Report. 2006.

B. J. Wielinga, A. T. Schreiber, J. A. Breuker. KADS: A Modelling Approach to Knowledge Engineering. In Knowledge Aquisition 4 (1), 1992.