A Study of Machine Reading from Multiple Texts

Peter Clark and John Thompson

Networked Systems Technology, Boeing Phantom Works, Seattle, WA 98124, USA
peter.e.clark@boeing.com, john.a.thompson@boeing.com

Abstract

A system that seeks to build a semantically coherent representation from multiple texts requires (at least) three things: a representation language that is sufficiently expressive to capture the information conveyed by the text; a natural language engine that can interpret text and generate semantic representations in that language with reasonable reliability; and a knowledge integration capability that can integrate information from different texts and from background knowledge into a coherent whole. In this paper we present a case study of these requirements for interpreting four different paragraphs of text (from different sources), each describing how a two-stroke combustion engine behaves. We identify the challenges involved in meeting these requirements and how they might be addressed. One key feature that emerges is the need for extensive background knowledge to guide the interpretation, disambiguate, and fill in gaps. The resulting contribution of this paper is a deeper understanding of the overall machine reading task.

Introduction

Our vision of a machine reading system is one which can process text so as to be able to answer questions concerning the text's subject matter in a coherent way, including about facts implied by but not explicitly stated in the text. Our assumption is that such a system will need to form a coherent, "deep" representation of the text's content in order to do this (as opposed to question-answering using shallow information retrieval methods). Despite its appeal, the challenges of building such a system remain formidable: Corpus-based techniques for knowledge extraction, e.g., (Schubert 2002, Banko and Etzioni, 2007), tend to accumulate noisy fragments of knowledge of rather restricted types rather than an integrated, coherent model of a text's subject matter, while full ("deep") semantic processing of text typically produces output containing numerous errors (Bos, 2008). Despite this, we believe machine reading is still an achievable goal: an obvious way forward is to have the machine read multiple texts about the same topic and then integrate the results together, using redundancy to identify the reliable portions of the interpretations and discard the unreliable portions. Our goal in this paper is to explore this hypothesis and the requirements for it through a case study.

To do this, we performed a manual analysis of the machine reading task for four paragraphs of text, shown in the appendix of this paper, describing the behavior of two-stroke engines, i.e., of the task of interpreting the individual texts and then integrating the separate representations together. We first ran each text through Boeing's Language Understanding Engine (BLUE) (Clark and Harrison, 2008), and then analyzed issues which arose in the individual interpretations, and what it would take to then integrate those interpretations together. This paper presents this analysis, which is largely a thought experiment on the machine reading task, with particular emphasis on the integration step. The resulting contribution of this paper is a deeper understanding of the requirements for machine reading, and a sketch of a possible future implementation.

The four texts were taken from the 2007 intermediate evaluation of the Mobius NLP system, following its earlier application for processing texts about a different topic (the human heart) (Barker et al, 2007). The texts themselves came from Web pages about two-stroke engines, and were slightly simplified for the Mobius project. Even in the simplified form, they still present formidable challenges for representation, interpretation, and integration.

To generate the interpretations of each paragraph, BLUE converts sentences into a set of Skolemized, ground first-order logic assertions, performing simple word sense disambiguation, semantic role labeling, coreference resolution, and structural transformation. Further details are given in (Clark and Harrison, 2008). For example, the interpretation of sentence 11 in the appendix is:

```
;;; 11. "Igniting the mixture causes an explosion"
 isa(mixture01,mixture#n1),
 isa(ignite01,light#v4),
 isa(explosion01,explosion#n1),
 object(ignite01,mixture01),
 causes(ignite01,explosion01).
```

where mixture01 (etc) are Skolem individuals and mixture#n1 (etc) are WordNet word senses.

Then, looking at BLUE's interpretations of the four paragraphs, we analyzed challenges that arose in three areas: (I) knowledge representation, (II) language interpretation of the individual paragraphs, and (III) knowledge integration of the individual paragraph interpretations into a coherent whole. The knowledge integration step was performed manually, as we do not yet have an implementation. A goal in the analysis was to identify the tasks involved in this step and the requirements for a future implementation.

Analysis

Part I: Knowledge Representation

1. Representation Language Requirements

(a) Representing Quantification

Like many "how things work" texts, our four paragraphs each describe a "typical" two-stroke engine, which BLUE translates to a set of Skolemized ground assertions about that "typical" engine. To "raise" these assertions to be about *all* two-stroke engines, BLUE then adds universal quantification over the Skolem variable denoting the topic of the paragraph (i.e., the two-stroke engine Skolem), and existentially quantifies over all other Skolem variables (denoting the piston, the cylinder, etc.). One can view the set of assertions as describing a prototypical two-stroke engine, in that the properties described are true for all two-stroke engines. This prototype-style representation has been used in other NLP systems also (e.g., Barker et al, 2007).

While this approach works works well (including for our four paragraphs), it breaks down for sentences which quantify over individuals other than the paragraph topic, and for sentences with more complex quantification patterns, i.e., that do more than relate specific individuals together. Examples of quantifying over non-topic individuals are: most generic sentences (e.g., "Burning oil produces pollution"), statements which generalize beyond the paragraph topic (e.g., "The spark plug contains an electrode" refers to spark plugs in general, not just spark plugs of two-stroke engines), and when the author discusses two topics together (e.g., "The two-stroke engine is simpler than the four-stroke engine."). To handle these, the system would need to treat the text as describing several different prototypes (rather than just one about the paragraph topic), and decide which sentence contributes to which prototype.

An example of a sentence with a more complex quantification pattern in our four texts is:

> 4. The two-stroke engine has one power stroke for every revolution of the crankshaft.

BLUE is currently unable to represent the semantics of this sentence, as it does not fit into a simple prototype (forall...exists...) quantification pattern. A fully capable machine reading system would need to address all these representational challenges.

(b) Representing Change

In any process, objects can change their location or other relationships to other objects. For example, in our texts the piston keeps moving and the resulting interpretation produces contradictory statements about the piston's location. Similarly, the fresh mixture of fuel and air is first moved into the crankcase region of the cylinder and later into the combustion chamber of the cylinder, e.g.,:

> 23. The vacuum in the crankcase sucks a fresh mixture of air-fuel-oil into the cylinder.

The representation and reasoning system must be able to correctly handle these changes of state that occur throughout the process, requiring additional representational machinery besides a static graph (set of assertions) about objects and their relationships, e.g., using situation calculus or event calculus. In addition, for sentence 23, the property of being "fresh" is a fluent, and will change once the mixture is ignited. BLUE is currently not able to represent change; again, a fully capable machine reading system would need to address this challenge.

(c) Representing second-order assertions

While most of the text can be expressed as first-order logic assertions, statements about purpose, goals, and justifications express relationships between propositions, requiring additional representational machinery. For example:

> 7. Two-stroke engines can operate in any orientation because the lubrication is in the cylinder.

Here the text is essentially presenting part of an argument or proof (alternatively, it could be considered an if-then rule), requiring second-order machinery. BLUE has a limited ability to represent second-order expressions (by placing propositions as arguments to propositions), but is not able to manipulate them in any sensible fashion, again a challenge for a full machine reading system.

2. Ontology Requirements

BLUE uses WordNet as its ontology of concepts, and the University of Texas (UT)'s slot dictionary[1] as its ontology of semantic relations (predicates). While these are reasonable starting points, they are not the complete story for machine reading. Although WordNet has vast coverage and extensive word-to-concept knowledge, it also has several well-known problems, in particular: sometimes sense distinctions are hard even for a human to discern (inter-annotator agreement for human taggers can be low); the hypernym tree does not always encode a subsumption relation (Kaplan and Schubert, 2001); and it has limited additional world knowledge besides the hypernym and part-of hierarchies. In addition, a machine reading system needs to be able to expand the ontology it is using to add new domain-specific concepts, including multiword concepts (e.g., for "reed valve", or "two-stroke engine") and concepts not already in WordNet (e.g., "biodiesel").

The choice of semantic relation vocabulary is also a difficult one. Typically, either the relational vocabulary is very big (making selection of the right semantic relation hard), or the relations are heavily overloaded to mean different things in different contexts (making their

[1] http://www.cs.utexas.edu/users/mfkb/RKF/tree/components/specs/slotdictionary.html

semantics unclear). UT's slot dictionary is between these two extremes, providing a working vocabulary of about 100 relations. This vocabulary was generally adequate, although in several cases shoehorning a relationship into this vocabulary seems to lose information (e.g., the prepositions in "through the pipe", "along the road", "under the bridge" all seem to map to the semantic role "path", apparently losing distinctions suggested by the different prepositions), and in some cases the appropriate semantic relation is unclear (e.g., what is the appropriate relation between "power" and "stroke" in "power stroke"?). Again, a good relational vocabulary is an important requirement for a machine reading system.

Part II: Language Interpretation

1. Explicit Knowledge: Basic Language Processing

As is well known, the basic task of language interpretation - here, synthesizing Skolemized logic assertions from text - is formidable due to the many sources of ambiguity in language, the many ways an idea can be expressed, and the imprecision and incompleteness of normal human communication. Common language interpretation challenges include: structural ambiguity (e.g., prepositional phrase attachment), word sense disambiguation, semantic role labeling, coordination, reference and anaphora resolution, generics and quantification, time, plurals and collectives, comparatives, implicit arguments, proper names, negation, modals, and pragmatics and discourse structure. As these have been extensively discussed in the NLP literature, we only highlight some interesting examples which arose in our texts below.

5. The power stroke starts when the spark plug emits a spark.

An interesting word sense disambiguation challenge here is whether the "spark" is an object or an event; one could argue that a spark (like an arc or a lightening bolt) is not a physical object in the normal sense, but rather is a short process (event). Also in this sentence, we would want to interpret "starts" as a temporal relation, rather than (as BLUE does) a verb.

6. Two-stroke oil is mixed with fuel to provide lubrication.
7. ...the lubrication is in the cylinder...

Sentence 6 presents another interesting word sense disambiguation challenge as to whether "lubrication" is a substance or a lubricating event. Treating it as an event may seem slightly more natural, but then sentence 7 treats lubrication as a substance (a mixture consisting of the oil and the fuel). One could argue that "lubrication" in 6 should therefore also be a substance; or alternatively if it is an event there then the text has violated the "one sense per text" assumption that is commonly made in word sense disambiguation. Again, an automated system would need to unravel these difficult choices.

9. At the begining of the combustion stroke,....

In this sentence, the fronted prepositional phrase needs to be recognized as a relative temporal reference rather than about a "beginning" object.

13. The piston compresses the mixture in the crankcase as it moves down.

In 13, resolving the anaphoric referent for “it” is challenging, as the commonly used default rule of "it = the most recent noun" does not work in this case. Instead, domain knowledge or knowledge from elsewhere in the text is required to realize that the piston is the thing which is moving.

23. The vacuum in the crankcase sucks a fresh mixture of air-fuel-oil into the cylinder.

BLUE interprets "air-fuel-oil" as a single token, thus losing information that the mixture is a combination of air, fuel, and oil.

31. The crankcase is the part of the cylinder on the other side of the piston.

Sentence 31 has attachment ambiguity: is it the crankcase (correct) or the cylinder (incorrect) that is on the other side of the piston? (BLUE got this wrong). Knowledge that the the piston is in the cylinder, and thus the cylinder cannot be on the other side of the piston, could help resolve this.

35. The sides of the piston act like valves.

This sentence is a vague and complex reference to the fact that the piston covers and uncovers different holes (ports) as it moves, allowing gas to move in and out of the cylinder. If we generate a naive, literal interpretation of the text, then what semantic relation does "like" refer to? (BLUE just defaults to "related-to").

2. Implicit Knowledge: Pragmatics

One might expect that at least the basic knowledge about a two-stroke engine, e.g., its parts or behavioral event sequence, are stated often and explicitly. In fact, this knowledge is rarely stated explicitly: the engine’s parts are mentioned but rarely declared explicitly as parts of the engine; events are stated but rarely declared explicitly as part of the engine's behavior, and their ordering is rarely made explicit. Rather, the authors are relying both on pragmatic knowledge (e.g., assume physical objects are part of the entity being described) and background knowledge (e.g., the reader already knows that pistons are engine components) to convey information.

(a) Part-Of Relations

As part-of relations are rarely stated explicitly, we assumed in our analysis that each solid object mentioned was a part of the two-stroke engine. (Background knowledge is needed to identify whether an object is solid or fluid). The fluids, however, are not assumed to be parts of the engine. Using this heuristic rule, parts such as the cylinder, piston, and spark plug are identified as parts of the engine.

Sometimes this assumption does not always hold, or is at least questionable. For example,

> 24. The air-fuel-oil mixture is sucked from the carburetor.

suggests the carburetor is part of the engine, although this is arguable. More generally, less focused texts can easily mention other objects e.g., the car chassis or the driver, which should not be treated as part of the engine. Clearly more sophisticated heuristics for capturing this pragmatic knowledge are needed.

(b) Event Sequences and Temporal Relations

A key issue for these "how things work" texts is how to deduce the sequential and temporal relations among the events within a sentence and across adjacent sentences. Often such relations are vaguely stated or unstated (again relying on pragmatics to recover them). For example, in:

> 8. A two-stroke engine's combustion stroke occurs when the spark plug fires.

what are the semantics of "occurs when"? The most we can say from a naive reading of the English is that there is some temporal overlap between the combustion stroke and the firing. But from common sense about engines, we know that the firing of the spark plug is a short event, and from pragmatics we know the text is generally describing sub-events of the combusion stroke, and thus the likely intended relation is a "starts-when" connection.

More commonly, the temporal order is simply not stated:

> 29. Igniting the mixture creates energy.
> 30. The combustion chamber captures the energy.

In general, pragmatics suggests that the author will describe events in order, and so the system can assume the capturing follows the igniting. This heuristic regarding consecutive sentences works well in general, but there are exceptions that have to be handled too.

(c) Causal Relations

Besides temporal relations between events, there may also be causal relations. For example:

> 12. The explosion forces the piston down.

Here "force" can be considered a causal relation between the exploding and the (unstated) moving event (an NLP engine would need to fill in this missing element). More generally, however, the causal links between events are unstated and would need to be inserted using background knowledge. For example:

> 10. This mixture ignites when the spark plug generates a spark.

implies a causal relation, although only a temporal relation is stated. Again, commonsense knowledge that sparks can trigger explosions is needed to fill in this connection. A possible source of such knowledge might be from work by Schubert (2002), whereby commonsense statements of possibility (e.g., "airplanes can fly", "people can eat food") are extracted automatically from text.

In general, in an artifact each part is there for a reason, and there is a purposeful rationale for the events in its behavior (the spark plug emits a spark in order to ignite the gas; the gas is ignited in order to create pressure; etc.). Understanding this rationale (teleology) is critical to understanding the artifact, however it is rarely stated explicitly, and requires pragmatic knowledge about the structure of the text and background knowledge to recover it.

Part III: Knowledge Integration

Aligning Texts

Perhaps the most important task for machine reading is knowledge integration: combining information from different texts and from background knowledge to form a coherent whole. Given that interpretations for individual paragraphs are likely to be incomplete and erroneous in many places, exploiting redundancy and background knowledge is key to removing errors, filling in gaps, and reinforcing correct statements. Using our Skolemized predicate representation, this process can be visualized as a sophisticated graph integration (and editing) operation, where the four graphs denoting the interpretations of the four texts are combined. While we have not implemented this operation, we here analyze what it involves in the context of our four texts.

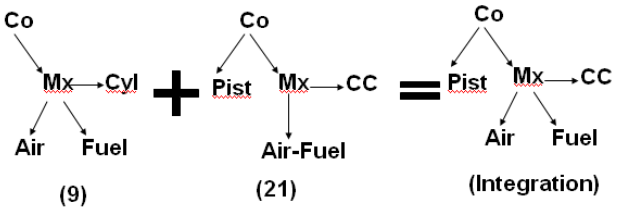
In the rest of this section we use graphs to show the structure of the NL interpretations and how they can be combined. For presentation purposes, instances (nodes) in the graphs have abbreviated names (e.g., Co = "compress"), and relations (arcs) are not named. The full names should be fairly clear (in most cases) from the quoted source texts.

A basic operation for integrating representations is to identify and unify (equate) coreferential individuals in the different texts' interpretations. For example, the piston in each of the four representations might be assumed to denote the same piston, and hence the Skolems denoting them unified (equated). However, care must be taken as sometimes there are multiple objects of the same type in a text (e.g., multiple "pieces" of mixture, multiple compress events), and so matching must also take into account the relations that each individual participates in. In addition, world knowledge is often needed to relate one representational fragment from one text with another fragment from another, e.g., when one fragment implies another, or is logically equivalent to the other. Below we provide some examples of these issues in these texts and discuss the issues involved in implementing them.

As a first example, text 2 and text 3 both describe the compression of the air-fuel mixture:

> 9. ...the mixture of fuel and air in the cylinder has been compressed.

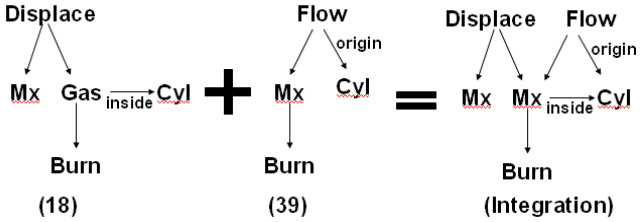
21. The piston compresses the air-fuel mixture in the combustion chamber.



In this case alignment is relatively straightforward, but with two complications. First, the location of the mixture is described at different levels of granularity (in the cylinder (9), vs. in the combustion chamber (21)). If the system had knowledge that the combustion chamber is part of the cylinder, it could align these two nodes, retaining the most specific one (combustion chamber). Such knowledge might come from background knowledge, or from other texts. In this case, text 4 states this piece of knowledge in sentence 28. Alternatively, the system could hypothesize a part-subpart relation given this alignment. Second, the system needs to realize that "the mixture of fuel and air" (9) is coreferential with "the air-fuel mixture" (21). Although it is reasonable for the system to assume this (as they are both mixtures and are related to the compressing event in the same way), our NLP system interpreted "air-fuel" as a single token, and thus the integrator would have difficulty realizing "air-fuel" and "fuel and air" were equivalent.

As a second example, texts 2 and 4 both describe expulsion of the burnt gas:

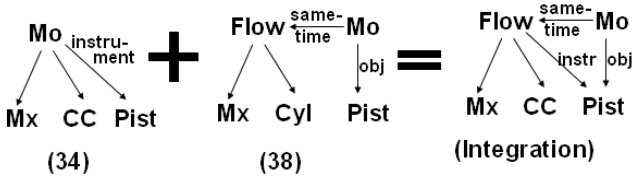
18. The mixture displaces the burned gases in the cylinder.
39. Burnt mixture flows out of the cylinder...



The integration here is not completely straightforward, because there are in fact two mixtures involved (one mixture displaces the other), and the "burnt mixture" in (39) is coreferential with "burned gases" not "the mixture" in (18). Thus naively assuming coreference based on same name/same concept would cause misalignment in this case. To achieve the correct alignment, the system would need to also match the property of being burnt. In addition, there is domain knowledge which can be used to reinforce this alignment, namely that: if a fluid is displaced from a container (e.g., the gases in 18), then it was inside the container and will flow out of the container (e.g., the mixture in 39). This piece of background knowledge would support the alignment of gases (18) and mixture (39), and also add a causal link that the displacement causes the flowing.

A third example of alignment is between sentences 34 and 38 (although these are both from text 4, they exhibit the same integration challenges as if they were from different texts). Both describe the piston moving the mixture:

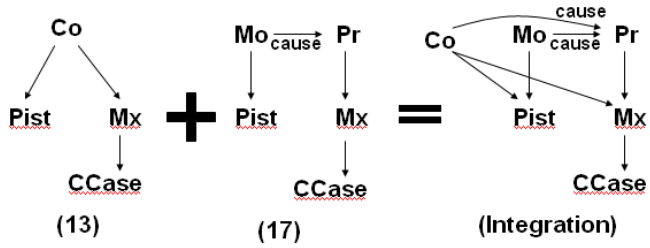
34. The piston moves mixture into the combustion chamber.
38. ...the mixture to flow into the cylinder as the piston moves.



The two moves here are different: the first is transitive and the second is intransitive. Again world knowledge can be used to identify the relationship between them and make the alignment, namely by using the rule IF X moves Y THEN (typically) X moves AND Y moves. In this case, as the piston moves the mixture, the piston moves (aligning with the move in 38) and the mixture moves (aligning with the flow in 38, given knowledge that a flow is a type of move), resulting in alignment.

A fourth example is sentences 13 and 17 in text 2 describing compression of the mixture in the crankcase:

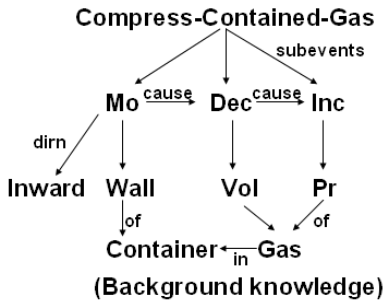
13. The piston compresses the mixture in the crankcase...
17. The piston's movement pressurizes the mixture in the crankcase.



Again, to align these, the system needs background knowledge of the relationship between compressing and pressurizing, i.e., that compressing a gas causes pressurizing of the gas.

In addition to using simple background knowledge as illustrated above, a knowledge integration system would greatly benefit from having larger script-like expectations about stereotypical events in the world. These expectations would allow gaps in the textual knowledge to be filled in, and potentially allow errors in the textual knowledge (or its interpretation) to be corrected. For example, one general script relevant to two-stroke engines is "compressing a contained gas", which might be encoded as:

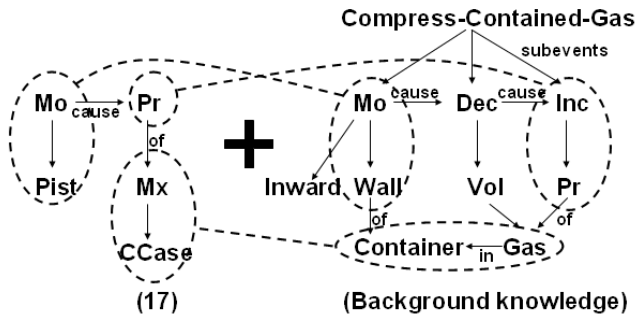
> A wall of the container is moved inwards, causing a decrease in volume of the gas, causing an increase in pressure of the gas.



This script can be aligned with sentence 17:

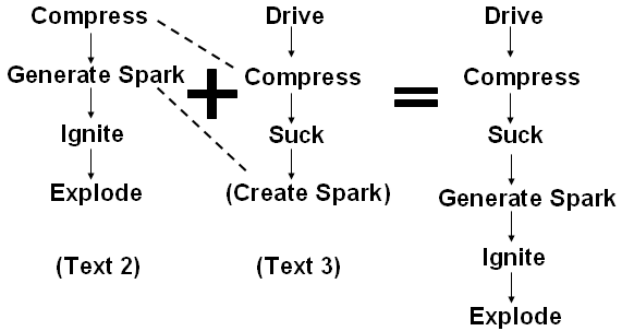
17. The piston's movement pressurizes the mixture in the crankcase.

given additional knowledge that "pressurize" means "to increase in pressure", and that IF X causes Y AND Y causes Z THEN (typically) X causes Z:



In the above, through graph matching the moving piston is aligned with the moving wall, pressurize is aligned with increasing pressure, and the mixture in the crankcase is aligned with the gas in the container. Given this alignment, we not only validate the interpretation's structure but we now have additional inferences available to us, e.g., the volume of the mixture is decreased; and the piston is one of the walls of the crankcase. Thus this is very powerful, but of course requires availability of a library of general scripts like this in the first place.

Finally, as described earlier, each text produces an event sequence which needs to be aligned with the others. For example, text 2's sequence includes a compress (9), generate (10), ignite (11), and explode (11), and text 3's sequence includes a drive (move) (20), compress (21), suck (23), and at the end an implicit creation of a spark (26):



In this case, the two compress events match straighforwardly, but the spark generation event (10) is only implicitly referred in text 3:

26. The spark from the spark plug begins...

Unfortunately text 3 does not explicitly say that the spark is generated, just that it is "from the spark plug," making alignment difficult. However, if the system had formalized background knowledge about spark plugs (their behavior is to create sparks, sparks are short-lived), it could infer that the existence of a spark from a spark plug implied the creation of that spark from the plug, hence creating a second alignment point between the two event sequences. In general, as two event sequences are merged, the system would need additional background knowledge to identify the correct alignments when there was ambiguity, e.g., A+B+C & A+D+C becomes A+B+D+C if B precedes or causes D, or becomes A+B+C if D is a subevent of B.

To summarize, the tasks we have identified and used for knowledge integration are:

- coreference identification (same name/type/relation to other instances)
- "logical" transformations (e.g., in appropriate contexts: part and subpart, event and subevent, distant cause and immediate cause). Yeh et al. (2003) and Tatu (2007) have both compiled catalogs of such transformations.
- inference rule application (e.g., compress gas → pressurize gas)
- matching with scripts

In addition, as integration is a search process, a metric for the "quality" of the integrated representation is needed to guide the search (e.g., degree to which the representation accounts for information in the individual texts; degree of consistency with background knowledge; internal consistency and completeness).

Missing Knowledge

Even with four paragraphs of text, there is still knowledge missing in the (hand-built) integrated representation. In some cases, the missing knowledge could be acquired by using more texts, or performing "targeted reading" to find a specific fact. In other cases, general knowledge can be used to fill in the gaps. Some examples are as follows:

1. Where is the spark plug? Nothing in the texts say that it is spatially connected to anything. It is possible that another text may answer this question. (It is worth noting that "how things work" texts are often accompanied by diagrams that (we assume) are inaccessible to the computer, and so the computer is at a disadvantage working from texts alone).

2. What causes the ignition? Although sentence 10 says:

   10. This mixture ignites when the spark plug generates a spark.

   the text does not say that the spark is the *cause* of the ignition, or that the spark is in contact with the

mixture. This is commonsense knowledge about gas explosions, which is less likely to be spelled out explicitly. If the computer had background knowledge of the "gas explosion" script (e.g., ignite → explode → pressurize → expand), this missing knowledge could be filled in. This would also help answer the earlier question of where the spark plug is: If the spark plug emits a spark (sentences 5,10), and the spark is in contact with the mixture (gas explosion script), then the plug is also (likely) in contact with the mixture.

3. What happened to the fresh mixture after the explosion, and where did the burned gas appear from? Although we know the mixture becomes the burned gas, this is not stated in the text and from the computer's point of view the burned gas magically appears and the fresh mixture remains after the explosion. Again, background knowledge about explosions is needed to fill in this unstated detail.

4. There is no mention of the passageway between the combustion chamber and the crankcase. Further texts could provide this. Again, the computer is handicapped without the ability to process diagrams.

5. How do the intake and exhaust ports get covered and uncovered? In a two-stroke engine the moving piston covers and uncovers the ports, but the text only states:

   > 14. As the piston approaches the bottom of its stroke, the exhaust port is uncovered.
   > 16. As the piston reaches the bottom of the cylinder, the intake port is uncovered.

   These describe the time of the uncovering but not that the piston plays the role as the cover. Also, the event of *covering* the ports is not mentioned, although commonsense tells us that if they are uncovered, then they must have been covered previously.

From these examples, it is clear that to construct a more complete model of a two-stroke engine, the system would need both more text (in particular for the spatial relationships) and more background knowledge (for some of the "obvious" unstated facts). In particular, basic knowledge/scripts about gas compression, gases exploding, fluid movement, and valves would be very helpful for filling in details and helping disambiguate the text.

Incorrect and Conflicting Knowledge

So far we have largely assumed correct and consistent text and its interpretation. However, in practice there will be errors in the interpretations, and a machine reading system will need to tolerate them. Sources of error include:

1. Errors in language interpretation (e.g., wrong PP attachment). As discussed earlier, even with today's state-of-the-art NLP systems, there are frequently errors in the interpretations.

2. Variants in the artifact being discussed. Not all two-stroke engines are the same, for example a two-stroke engine in an ocean liner is quite different to one in a lawnmower. In addition, our four texts assume an orientation (that "up" is towards the combustion chamber), but other texts assume a different orientation. Given multiple texts, there may be genuine differences in the item being described, causing conflicts when trying to integrate the knowledge. For this reason, it is important to work with texts that are as homogeneous as possible.

3. Different levels of sophistication, e.g., grade school vs college level descriptions. Related to the earlier point, authors may simplify to convey an idea, another potential source of conflicting knowledge.

4. The author may simply make incorrect statements, e.g., through loose use of language or lack of understanding. For example, another text we looked at stated "The spark plug ignites a spark.", which conflicts with a normal understanding of ignition.

For machine reading, it is thus critical to select texts that are as homogeneous as possible, and that use simple (rather than "flowery") language to ease NL interpretation. Homogeneity might be measured by overlap of words, or overlap of technical vocabulary, in the texts. Simplicity might be measured by sentence length, parse tree depth, prepositional phrase density, or some vocabulary measures.

Sources of World Knowledge

Throughout this paper we have cited the need for background knowledge. Where might this knowledge come from? Four potential sources are:

1. Existing resources: e.g., WordNet, VerbNet, FrameNet, Cyc, The Component Library, OntoSem.

2. Automatic acquisition from text: Numerous corpus-based methods exist for acquiring knowledge from text, e.g., taxonomic knowledge, partonomic knowledge, paraphrases.

3. Web volunteers: Although producing rather noisy data to date, there is significant potential in using communities of volunteers to acquire knowledge through on-line acquisition "games", e.g., OpenMind.

4. Manual encoding: Although expensive, the task of manually encoding knowledge can be bounded by aiming just for general, highly reusable concepts rather than all world knowledge.

Summary and Lessons Learned

Although more experiments and analyses are needed, there are many interesting lessons that can be drawn from this exercise, in particular:

1. Change needs to be represented and reasoned about. For "how things work" texts, it is important to represent and reason about change, and not just work with a static first-order logic representation. The

underlying representation produced by our language interpreter BLUE is inadequate in this regard.

2. Implicit knowledge needs to be recovered. There is a surprisingly large amount of implicit knowledge in text. Some of this (e.g., engine parts, event ordering) requires heuristics about pragmatics and discourse to recover. Other parts require basic commonsense knowledge to fill in the "obvious" gaps.
3. Four paragraphs is not enough. Even with perfect interpretation, the four paragraphs we are working with provide a rather incomplete description of the two-stroke engine. Working with a larger set (e.g., 20-100 texts) would provide more completeness, although also increase the integration challenges.
4. Background knowledge plays a critical role. (This includes having a good starting ontology and semantic relation vocabulary). In our analysis, background knowledge was potentially useful for almost every disambiguation and integration decision being made.
5. Knowledge integration is challenging. We have presented several examples of aligning and integrating interpretations, but further work is needed to distill this into a formal algorithm.

Despite these challenges, the algorithmic nature of the steps discussed in knowledge integration is encouraging. We remain hopeful that the process can be automated, and that machine reading is an achievable goal in the near future.

References

Banko M., Etzioni, O., 2007. Strategies for Lifelong Knowledge Extraction from the Web. In *Proc. Fourth Int. Conf. on Knowledge Capture (KCAP) 2007.*

Barker et al. (2007). Learning by Reading: A Prototype System and Lessons Learned. *Proc. AAAI 2007*

Bos, J. 2008. Introduction to the Shared Task on Comparing Semantic Representations. *Proc. STEP 2008.* (also see subsequent articles in the proceedings).

Clark, P., Harrison, P. 2008. “Boeing's NLP Sys-tem and the Challenges of Semantic Representation”, *STEP 2008.*

Kaplan, A. N., and Schubert, L. 2001. *Measuring and Improving the Quality of World Knowledge extracted from WordNet.* Technical Report, Univ Rochester.

Schubert, L. 2002. Can we derive general world knowledge from texts?", M. Marcus (ed.), *Proc. HLT 2002*

Tatu, M. 2007. *Intentions in text and semantic calculus.* PhD Thesis, UT Dallas.

Yeh, P, Porter, B., Barker, K. 2003. Using Transformations to Improve Semantic Matching. In *Proc KCap'03*.

Appendix: The Four Paragraphs

TEXT 1

1. Two-stroke engines are powerful devices.
2. Two-stroke engines are also lightweight devices.
3. Two-stroke engines are used for handheld devices that require a lot of power in a lightweight configuration.
4. The two-stroke engine has one power stroke for every revolution of the crankshaft.
5. The power stroke starts when the spark plug emits a spark.
6. Two-stroke oil is mixed with fuel to provide lubrication.
7. Two-stroke engines can operate in any orientation because the lubrication is in the cylinder with the fuel and the piston.

TEXT 2

8. A two-stroke engine's combustion stroke occurs when the spark plug fires.
9. At the beginning of the combustion stroke, the mixture of fuel and air in the cylinder has been compressed.
10. This mixture ignites when the spark plug generates a spark.
11. Igniting the mixture causes an explosion.
12. The explosion forces the piston down.
13. The piston compresses the mixture in the crankcase as it moves down.
14. As the piston approaches the bottom of its stroke, the exhaust port is uncovered.
15. The pressure in the cylinder forces exhaust gases out of the cylinder.
16. As the piston reaches the bottom of the cylinder, the intake port is uncovered.
17. The piston's movement pressurizes the mixture in the crankcase.
18. The mixture displaces the burned gases in the cylinder.

TEXT 3

19. In a two-stoke engine there is a compression stroke, followed by a combustion stroke.
20. The compression stroke occurs when the crankshaft's momentum drives the piston up in the cylinder.
21. The piston compresses the air-fuel mixture in the combustion chamber. 22. A vacuum is created in the crankcase.
23. The vacuum in the crankcase sucks a fresh mixture of air-fuel-oil into the cylinder.
24. The air-fuel-oil mixture is sucked from the carburetor.
25. There is a reed valve between the carburetor and the cylinder.
26. The spark from the spark plug begins the combustion stroke when the piston reaches the top of the cylinder.

TEXT 4

27. In a two-stroke engine, one side of the piston is also a side of the combustion chamber.
28. The combustion chamber is the part of the cylinder where the air-fuel mixture is compressed.
29. Igniting the mixture creates energy.
30. The combustion chamber captures the energy.
31. The crankcase is the part of the cylinder on the other side of the piston. 32. The piston creates a vacuum in the crankcase.
33. The vacuum sucks in mixture from the carburetor through the reed valve. 34. The piston moves mixture into the combustion chamber. 35. The sides of the piston act like valves.
36. The cylinder has an exhaust port.
37. The cylinder also has an intake port.
38. The intake port allows mixture to flow into the cylinder as the piston moves.
39. Burnt mixture flows out of the cylinder through the exhaust port.