A Knowledge-Driven Approach to Text Meaning Processing

Peter Clark, Phil Harrison, John Thompson
Boeing Mathematics and Computing Technology
{peter.e.clark,john.a.thompson,philip.harrison}@boeing.com

Abstract

Our goal is to be able to answer questions about text that go beyond facts explicitly stated in the text, a task which inherently requires extracting a "deep" level of meaning from that text. Our approach treats meaning processing fundamentally as a modeling activity, in which a knowledge base of common-sense expectations guides interpretation of text, and text suggests which parts of the knowledge base might be relevant. In this paper, we describe our ongoing investigations to develop this approach into a usable method for meaning processing.

1 Overview

Our goal is to be able to answer questions about text that go beyond facts explicitly stated in the text, a task which, we believe, requires extracting a "deep" level of meaning from the text. We treat the process of identifying the meaning of text to be one of constructing a *situation-specific representation* of the scenario that the text is describing. Elements of the representation will denote objects and relationships in that scenario, some of which may not have been explicitly mentioned in the text itself. The degree to which a computer has "acquired the meaning" of some text will be reflected by its ability to answer questions about the scenario that the text describes.

A significant challenge for meaning processing is that much of the content of the target representation may not be explicitly mentioned in the text itself. For example, the sentence:

> (1) "China launched a meteorological satellite into orbit Wednesday."

suggests to a human reader that (among other things) there was a rocket launch; China probably owns the satellite; the satellite is for monitoring weather; the orbit is around Earth; etc. A system that has adequately "understood" the meaning of this sentence should include such plausible implications in its representation, and thus (for example) be able to identify this sentence as relevant to a query for "rocket launch events". However, none of these facts are explicitly mentioned in the text. Rather, much of the scenario representation needs to come from strong, prior expectations about the way the world might be, and meaning processing involves matching, combining, and instantiating these prior expectations with knowledge explicitly stated in text. Viewed this way, understanding is fundamentally a modeling process, in which prior knowledge and knowledge from text interact: Text suggests which scenarios in the knowledge base might be relevant; and scenarios from the knowledge base suggest ways of interpretating and disambiguating text.

This style of approach to meaning processing used to be popular in the 1970's and 1980's, e.g., (Cullingford, 1977; DeJong, 1979; Schank and Abelson, 1977), but has largely been abandoned for a number of reasons, both theoretical and pragmatic. Challenges include: the cost of building the knowledge base of expectations in the first place; controlling the matching process in a robust way; basic issues of knowledge representation (defining what the target should be in the first place); and the recent successes of knowledge-poor statistical approaches on certain classes of information retrieval tasks. However, despite these challenges, many question-answering tasks will remain out of reach of current systems until deeper representations of meaning are employed, and thus we consider that these challenges are important to address, rather than avoid.

In an earlier project (Clark et al., 2002), we explored methods for interpreting sentences about aircraft, expressed in a simplified version of English, using this knowledge-driven style of processing. Interpretation was performed by matching the sentences' NL-produced "logical forms" against pre-built representations of aircraft components and systems. Although effective for certain texts, the generality of this method was constrained in two ways. First, for successful matching, the approach requires the logical form of the input text to be

```
(launch_a_satellite_v1 has
  (superclasses (launch_v1 transport_v1)))      ; hypernyms

(every launch_a_satellite_v1 has
  (step_n1 ((a countdown_n1 with
              (location_n1 ((the location_n1 of Self)))
              (event_n1 ((the fly_v1 step_n1 of Self)))
              (before_r1 ((the fly_v1 step_n1 of Self)))
            (a fly_v1 with
              (vehicle_n1 ((the vehicle_n1 of Self))))))
   (vehicle_n1 ((a rocket_n1)))
   (agent_n1 ((a causal_agent_n1)))
   (cargo_n1 ((a satellite_n1)))
   (location_n1 ((a launchpad_n1))))
```

Figure 1: The representation (simplified) of the scenario "launching a satellite" in the knowledge-base, encoded in the language KM. (See the body of this paper for a summary of the semantics).

both fairly simple and fairly close, structurally, to the target matching structure in the knowledge base. Second, the cost of producing the knowledge base by hand is expensive, and the approach is limited to just those areas that the knowledge base covers.

To address these challenges, we are currently exploring a modified approach, inspired by Schubert's recent work on extracting common-sense knowledge from text (Schubert, 2002). Before building the full "logical forms" from text, which can be large and complex, and may require certain disambiguation commitments to be made prematurely, we are first extracting shorter fragments of information from text, and using these for matching against the knowledge base. In the simplest form, these fragments are simple subject-verb-object relationships, e.g., from

> (2) "Yesterday, Russia launched a spaceship carrying equipment for the International Space Station."

the system would extract the fragments:

```
("Russia" "launch" "spaceship")
("spaceship" "carry" "equipment")
```

In a more sophisticated form, the fragments also include prepositional phrases, e.g., from

> (3) "Alan applied for a job."

the system would extract the fragment:

```
("Alan" "apply" "" ("for" "job"))
```

These structures are essentially snippets of the full logical form, except that (i) they are simplified (some details removed), and (ii) many semantic decisions, e.g., word sense disambiguation, the semantic relationships between the objects, have been deferred until knowledge-based matching time. The task then, given several such fragments extracted from text, is to find the scenario in the knowledge-base that best matches these fragments, i.e., that can account for as many as possible. Through the matching process, many of the deferred disambiguation decisions are made.

Although the fragment representation is impoverished compared with the full logical form, our conjecture is that it still contains enough information to identify the core meaning of the text, in terms of identifying and instantiating the relevant scenario in the knowledge base, while simplifying the meaning processing task. We are thus seeking a "middle ground" between superficial analysis of the text and full-blown natural language processing. In some cases, including those we have examined, the scenario from the knowledge base, instantiated with fragments, is sufficient to answer questions about the text, with no further processing being needed. However in other cases, we may need to add a "second pass" in which a more computationally intensive matching process is then used to match the text's full logical form with the fragment-selected knowledge base scenario. This is still an area of investigation.

In addition, these fragments may form the basis for helping construct the knowledge base in the first place (Schubert, 2002). By processing a large corpus of text, we can automatically generate a large number of fragments that can then provide the "raw material" for a person to construct the scenario models from. Our conjecture is that knowledge acquisition will be substantially faster when treated as a process of filtering and assembling fragments, rather than one of authoring facts by hand from scratch. We describe our initial explorations in this direction shortly.

2 The Knowledge Base

We have recently been working with text describing various kinds of "launch" events (launching satellites, products, Web sites, ships, etc.). We describe our ongoing implementation of the above approach in the context of these texts.

2.1 Architecture

We envisage that, ultimately, the knowledge base (KB) will comprise a small number of abstract, core representations (e.g., movement, transportation, conversion, production, containment), along with a large number of detailed scenario representations. We anticipate that the former will have to be built by hand, while the latter can be acquired semi-automatically using a combination of text analysis and human filtering/assembling of fragments resulting from that analysis. At present, however, we are building both the core and detailed representations by hand, as a first step towards this goal.

Each scenario representation contains a set of axioms describing the objects involved in the scenario, the events and subevents involved, and their relationships to each other. Before describing these in more detail, however, we first describe the KB's ontology (conceptual vocabulary).

2.2 The Ontology: Concepts

We are using WordNet (Miller et al., 1993) as the starting point for the KB's ontology. Although WordNet has limitations, it provides both an extensive taxonomy of concepts (synsets) and a rich mapping from those concepts to words/phrases that may be used to refer to them in text. This provides useful knowledge both for identifying coreferences between different representations that are known to relate (e.g., between a representation of "launching" and a representation of "moving", where launching is defined as a type of moving), and also for matching scenario representations with text fragments when interpreting new text (Section 3.2). The use of WordNet may also make semi-automated construction of the scenario representations themselves easier, if the raw material for these representations is derived from text corpora. We are also adding new concepts where needed, in particular concepts that we wish to reify which are described by phrases rather than a single word (thus not in WordNet), e.g., "launch a satellite", and correcting apparent errors or omissions that we find.

As a naming convention, rather than identify a synset by its number we name it by concatenating the synset word most commonly used to refer to it (as specified by WordNet's tag statistics), its part of speech, and the WordNet sense of that word corresponds to that synset. For example, `bank_n1` is our friendly name for synset 106948080 (bank, the financial institution), as "bank" is the synset word most commonly used to refer to this synset, this synset is labeled with a noun part of speech, and "bank" sense 1 is synset 106948080. This renaming is a simple one-to-one mapping, and is purely cosmetic.

In WordNet, verbs and their nominalizations are always treated as (members of) separate concepts, although from an ontological standpoint, these often (we believe) refer to the same entity (of type event). Martin has made a similar observation (Martin, 2003). An example is a running event, which may be referred to in both "I ran" and "the run". To remove this apparent duplication, we use just the verb-based concept (synset) for these cases. Note that this phenomenon does not hold for all verbs; for some verbs, the nominalization may refer to the instrument (e.g., "hammer") used in the event, the object (e.g., "drink"), the result (e.g., "plot"), etc.

2.3 The Ontology: Relations

For constructing scenario representations, we distinguish between active (action-like) verbs and stative (state-like) verbs (e.g., "enter" vs. "contain"), the former being reified as individuals in their own right (Davidsonian style) with semantic roles attached, while the latter are treated as relations[1].

For events, we relate the (reified) events to the objects which participate in those events (the "participants") via semantic role-like relations (agent, instrument, employer, vehicle, etc.). We are following a fairly liberal approach to this: rather than confining ourselves to a small, fixed set of primitive relations, we are simply finding the WordNet concept that best describes the relationship. This is partly in anticipation of the representations eventually being built semi-automatically from text, when a similarly diverse set of relations will be present (based on whatever relation the text author happened to use). In addition, it simply seems to be the case (we believe) that the set of possible relationships is large, making it hard to work with a small, fixed set without either overloading or excessively generalizing the meaning of relationships in that set.

This eases the challenge that working with a constrained set of semantic roles poses, but at the expense of more work being required (by the reasoning engine) to determine coreference among representations. For example, if we use "giver" and "donor" (rather than "agent" and "agent", say) as roles in "give" and "donate" representations respectively, and "donate" is a kind of "give", it is then up to the inference engine to recognize that

[1] In practice, this separation of events and states is not always so clean at the boundaries: whether something is an event or state is partly subjective, depending on the viewpoint adopted, e.g., the level of temporal granularity chosen. For example "flight" can be considered an event or a state, depending on the time-scale of interest.

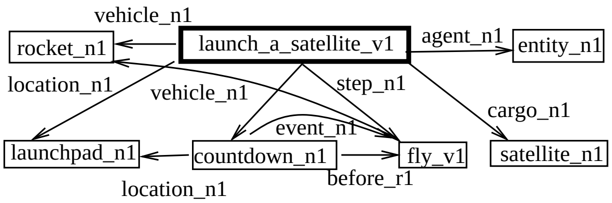

Figure 2: A graphical depiction of the "launching a satellite" scenario in the knowledge-base.

these probably refer to the same entity, which in turn requires additional world knowledge. We are currently using WordNet to provide this world knowledge. For example, in this case WordNet states that "donor" and "giver" are synonyms (in one synset), and hence the coreference can be recognized by the reasoning engine. In other cases one role concept may be a sub/supertype of the other.

This decision also means that we are using some WordNet concepts both as classes (types) and relations, thus strictly overloading these concepts. We are currently considering extending the naming convention to distinguish these.

2.4 Scenario Representations

The scenario representations themselves are constructed – currently by hand – by identifying the key "participants" (both objects and events) in the scenario, and then creating a graph of relationships that normally exist between those participants. In our example of "launching" scenarios, each type of launching (launching a satellite, launching a product, etc.) is represented as a different scenario in the knowledge base. These representations are encoded in the language KM (Clark and Porter, 1999), a frame-based knowledge representation language with well-defined first-order logic semantics, similar in style to KRL. For example, a (simplified) KM representation of "launching a satellite" is shown in Figure 1, and sketched in Figure 2. In the graphical depiction, the dark node denotes a universally quantified object, other nodes denote implied, existentially quantified objects, and arcs denote binary relations. The semantics of this structure are that: for every launching a satellite event, there exists a rocket, a launch site, a countdown event, ... etc., and the rocket is the vehicle of the launching a satellite, the launch site is the location of the launching a satellite, etc. The KB currently contains approximately 25 scenario representations similar to this.

These graphical representations are compositional in two important ways: First, through inheritance, a representation can be combined with representations of its generalizations (e.g., representations of "launching a satellite" and "placing something in position" can be combined). Second, different viewpoints/aspects of a concept such as launching a satellite are encoded as separate representational structures (e.g., the sequence of events; the temporal information; the spatial information; goal-oriented information). During text interpretation, only those representation(s) of aspects/views that the text itself refers to will be composed into the structure matched with the text.

3 Text Interpretation

3.1 Extraction of Knowledge Fragments from Text

Given the knowledge base of scenarios, our goal is to use it to interpret new text, by finding and instantiating the scenario in the KB which best matches the facts explicit in that text. To do this, first each sentence in the new text is parsed, and fragments are extracted from the parse tree. Parsing is done by SAPIR, a bottom-up chart parser used in Boeing (Holmback et al., 2000). Fragments are extracted by searching for subject-verb-object patterns in the parse tree, e.g., rooted at the main verb or in relative clauses. For example, given the sentence:

> (4) "A Russian Progress M-44 spaceship carrying equipment, food and fuel for the International Space Station was launched successfully Monday."

The fragments:

```
("" "launch" "spaceship")
("spaceship" "carry" "equipment")
("spaceship" "carry" "food")
("spaceship" "carry" "fuel")
```

are extracted. Note that at this stage word sense disambiguation has not been performed.

3.2 Matching Scenarios with Fragments

To match the scenario representations with the NLP-processed text fragments, the system searches for matches between objects in the representations and objects mentioned in the fragments; and relationships in the representations and relationships mentioned in the fragments. The subject-verb-object fragments are first broken up into two, e.g., `("China" "launch" "satellite")` becomes `("launch" "subject" "China")` and `("launch" "object" "satellite")` before matching. Then the system searches for a scenario representation where as many as possible word-syntacticrelation-word fragments match concept-semanticrelation-concept structures in the representation. Because we have used WordNet, each concept in the knowledge base has a set of associated words/phrases used to express it in English, and a word in a fragment "matches" a concept if that word is a member of these

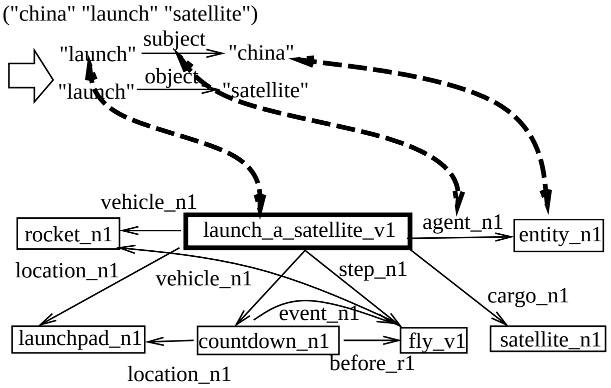

Figure 3: To interpret the text, the system finds the scenario representation that best matches the fragments extracted from the input text. Word sense and semantic role disambiguation is a side-effect, rather than a precursor to, this matching process.

associated words (i.e., the synset) for that concept (or one of its specializations or generalizations). This is illustrated in Figure 3. A simple scoring function is used to assess the degree of match, looking for the scenario with the maximum number of matching fragments, and in the case of a tie preferring the scenario with the maximum number of objects potentially matching some item in the text.

Note that it is only at this point that word sense and semantic relation disambiguation are performed. For example, in this case the fragments extracted from text best match the launch_a_satellite_v1 scenario; as a result, "launch" in the text will be taken to mean the launch_a_satellite_v1 concept (≈ word sense), as opposed to launching a product, launching a ship, etc.

One piece of information we are not currently exploiting in this matching process are the statistical probabilities that particular syntactic roles (grammatical functions) such as subject, direct object, etc., will correspond to particular semantic roles such as agent_n1, vehicle_n1, etc. These would help the matcher deal with ambiguous cases, where the current approach is not sufficient to determine the appropriate match. Automated methods for obtaining such statistics, such as (Gildea and Jurafsky, 2002), could be exploited for this task.

3.3 Question Answering

Having identified and instantiated the appropriate scenario representation in the knowledge base, that representation is now available for use in question-answering. This allows questions to be answered which go beyond facts explicitly mentioned in the text, but which are part of the scenario representation (e.g., a question about the rocket), and those requiring inference (using KM's inference engine, applied to the scenario and other knowledge in the knowledge base).

The inference engine currently requires questions to be posed in the native representation language (KM), rather than having a natural language front end. Given a query, KM will not just retrieve facts contained explicitly in the instantiated scenario representation, but also compute additional facts using standard reasoning mechanisms of inheritance and rule evaluation. For example, launch_a_satellite_v1 is a subclass of transport_v1, whose representation includes an axiom stating that during the move_v1 subevent, the cargo is inside the vehicle. Given an appropriate query, this axiom will be inherited to launch_a_satellite_v1, allowing the system to conclude that during the move subevent of the satellite launch – here fly_v1 – the satellite (cargo) will be inside the rocket (vehicle). The ability of the system to reach this kind of conclusion demonstrates, to a certain degree, that it has acquired at least some of the "deep" meaning of the text, as these conclusions go beyond the information contained in the original text itself.

4 Semi-Automatic Construction of the KB

For a broad coverage system, a large number of scenario representations will be necessary, more than can be feasibly built by hand. While fully automatic acquisition of these representations from text seems beyond the state of the art, we believe there is a middle ground in which the "raw material" for these representations can be extracted automatically from text, and which can then be rapidly filtered and assembled by a person.

As an initial exploration in this direction, we applied our "fragment extractor" to part of the Reuters corpus (Reuters, 2003) to obtain a database of 1.1 million subject-verb-object fragments. From this database, high-frequency patterns can then be searched for, providing possible material for incorporating into new scenario representations. For example, the database reveals (by looking at the various tuple frequencies) that satellites are most commonly built, launched, carried, and used; rockets most commonly carry satellites; Russia and rockets most commonly launch satellites; and that satellites most commonly transmit and broadcast. Similarly for the verb "launch", things which are most commonly launched (according to the database) are campaigns, services, funds, investigations, attacks, bonds, and satellites, suggesting a set of scenario representations which could then be built by searching further from these terms. Although these fragments are not yet assembled into larger scenario representations and word senses have not been disambiguated, further work in this direction may yield methods by which a user can rapidly find and assemble candidate elements of representations into larger structures, perhaps guided by the existing abstract models already in the knowledge base. Other corpus-based techniques such as (Lin and Pantel, 2001) could also be used to provide additional raw material for scenario construction.

5 Summary

We believe that text meaning processing, and subsequent question-answering about that text, is fundamentally a modeling activity, in which text suggests scenario models to use, and those models suggest ways of interpreting text. We have described some ongoing investigations were are conducting to develop this approach into a usable method for language processing. Although this approach is challenging for a number reasons, it offers significant potential for allowing question-answering to go beyond facts explicitly stated in the various text sources used.

References

Clark, P., Duncan, L., Holmback, H., Jenkins, T., and Thompson, J. (2002). A knowledge-rich approach to understanding text about aircraft systems. In *Proc. HLT-02*, pages 78–83.

Clark, P. and Porter, B. (1999). KM – the knowledge machine: Users manual. Technical report, AI Lab, Univ Texas at Austin. (http://www.cs.utexas.edu/users/mfkb/km.html).

Cullingford, R. E. (1977). Controlling inference in story understanding. In *IJCAI-77*, page 17.

DeJong, G. (1979). Prediction and substantiation: two processes that comprise understanding. In *IJCAI-79*, pages 217–222.

Gildea, D. and Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Holmback, H., Duncan, L., and Harrison, P. (2000). A word sense checking application for Simplified English. In *Third International Workshop on Controlled Language Applications*.

Lin, D. and Pantel, P. (2001). Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):343–360.

Martin, P. (2003). Correction and extension of wordnet 1.7. In *11th International Conference on Conceptual Structures (ICCS'03), to appear*.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. (1993). *Five Papers on WordNet*. Princeton Univ., NJ. (http://www.cogsci.princeton.edu/∼wn/).

Reuters (2003). Reuters corpus, volume 1, english language, 1996-08-20 to 1997-08019.

Schank, R. and Abelson, R. (1977). *Scripts, Plans, Goals and Understanding*. Erlbaum, Hillsdale, NJ.

Schubert, L. (2002). Can we derive general world knowledge from texts? In *Proc. HLT-02*, pages 84–87.