Using Episodic Memory in a Memory Based Parser to Assist Machine Reading

Kevin Livingston and Christopher K. Riesbeck

EECS Department
Northwestern University
2133 Sheridan Rd. Evanston, IL 60208, USA
{livingston; c-riesbeck}@northwestern.edu

Abstract

The central task for a Machine Reader is integrating information acquired from text with the machine's existing knowledge. Direct Memory Access Parsing (DMAP) is a machine reading approach that leverages existing knowledge and performs integration in the early stages of parsing natural language text. DMAP treats machine reading, fundamentally, as a task of knowledge recognition, and creates new knowledge structures and instances only when it cannot map the input text to existing knowledge. A goal of the research is to be able to use existing knowledge to improve the acquisition and integration of new knowledge, from (simplified) natural language. DMAP's understanding is driven by memory structures, and it maps immediately and directly to existing knowledge. This provides an opportunity to experiment with and evaluate methods for using existing knowledge (both semantic and episodic) to facilitate machine reading. We present the basic architecture of a DMAP implementation, three experiments to leverage existing episodic memory, and the implications of the experiments on future research.

Introduction

The central task for a Machine Reader is integrating information acquired from text with the machine's existing knowledge. Direct Memory Access Parsing (DMAP) is a machine reading approach that leverages existing knowledge and performs integration in the early stages of parsing natural language text. (Riesbeck and Martin 1985, Riesbeck 1986, Martin 1990) DMAP treats machine reading, fundamentally, as a task of knowledge recognition, and creates new knowledge structures and instances only when it cannot map the input text to existing knowledge.

DMAP uses a collection of language patterns that recursively map textual references to knowledge structures, and even specific instances in memory when possible. Instances include not only individuals like people, George W. Bush, or countries, Iraq, but also events in episodic memory, such as the Madrid train bombings of 2004.

This approach is different from those approaches evolving from the Natural Language Processing (NLP) tradition which are typically more bottom-up, performing in-depth syntactic parsing first (Calder et. al. 1988, Manning and Schutze 1999, Pease and Murray 2003). The result of the syntactic analysis is then translated into semantic structures, often with a one to one mapping between lexical and semantic forms. Those semantics may need further refinement, and are then ultimately integrated with the existing knowledge. Other language understanding techniques include primarily leveraging semantic memory such as Clark et. al. (2003) and techniques to extract specific facts from large scale source such as the web (Etzioni et. al 2004).

This paper discusses three ways in which existing knowledge, primarily episodic knowledge, can be leveraged to assist in machine reading. We present our experience with these techniques and how they affected the reading of a corpus of 183 short, single topic, texts written in simplified English. Vocabulary is not restricted, but a preference was made for simplified grammatical structure, as can be seen in the example story in the following section of this paper.

We discuss the investigation of three primary hypotheses. The first is that interpretations that produce references to existing knowledge (assertions in the KB) are better than interpretations that do not. The second hypothesis is that known instances in memory of individuals which are members of two sibling collections, can be used to allow coreferences to be established between those two collections while reading. For example, in the ResearchCyc (Lenat 1995) ontology there is no generalization link between the concept representing a bombing, and the concept representing terrorist attacks. However, DMAP will permit a reference to a bombing to be aligned to a reference to a terrorist attack, because in its episodic memory (Cyc) it knows of instances which are both bombings and terrorist attacks, and therefore has evidence that such a connection is possible. The third hypothesis was that semantic interpretations of a sentence that involved predicates that occur more frequently in the knowledge base with the predicates used in the interpretation for the previous sentences would provide better understanding of the input text.

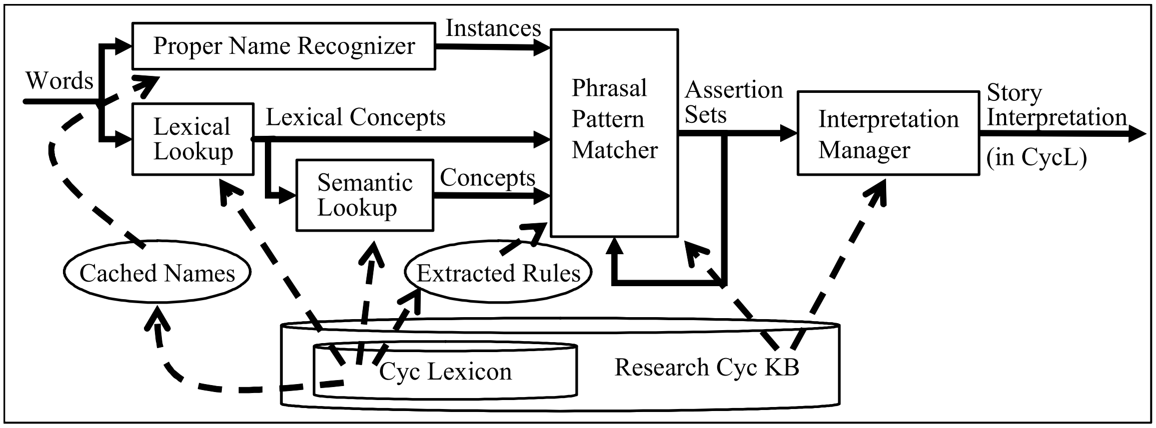

Figure 1: A functional depiction of the DMAP architecture. Boxes represent processes, and solid arrows the flow of data through the system. Dotted arrows represent interactions with the KB (memory) or cached portions of it.

To provide context for the discussion of these techniques we first present the architecture of our particular DMAP implementation, and provide a brief discussion of how the primary components operate.

Story Reading Example

This implementation of DMAP is built on top of the ResearchCyc knowledge base. The version of ResearchCyc used for these experiments contains over 1.2 million predicate logic assertions. (This is over two orders of magnitude larger than the frame based memories used with previous DMAP implementations.) The output of our implementation therefore is represented in CycL (the language of Cyc) and uses the representation and instances of the Research Cyc ontology and memory.

An example input to DMAP would be the following short story.

```
An attack occurred in Madrid.
The bombing killed 190 people.
The bombing was perpetrated by Al-Qaida.
```

Output from that story includes the retrieval of the following assertions, which already existed in memory.

```
(isa CityOfMadridSpain (CityInCountryFn Spain))
(isa CityOfMadridSpain CapitalCityOfRegion)
(isa 190 PositiveInteger)
(isa AlQaida TerroristGroup-Islamist)
(isa AlQaida InternationalOrganization)
(isa Person Collection)
(isa Spain Country)
(eventOccursAt
     TerroristAttack-September-8-2003-Madrid
     CityOfMadridSpain)
```

In addition DMAP proposed three additional assertions, which previously did not exist in memory, and it connected them to the existing representation of the event.

```
(perpetrator
     TerroristAttack-September-8-2003-Madrid
     AlQaida)
(deathToll
     TerroristAttack-September-8-2003-Madrid
     Person 190)
(isa TerroristAttack-September-8-2003-Madrid
     Bombing)
```

DMAP Prototype Architecture

Figure 1 shows the architecture of this version of DMAP. Words are translated into concepts in the Cyc knowledge base through three separate processes: proper name recognition, lexical lookup, and semantic denotation. The results of all of these processes are fed into the phrasal pattern matcher. Pattern matching produces sets of assertions representing possible interpretations for a given sentence. These assertion sets are combined with those of previous sentences and a preferred interpretation is produced by the interpretation manager to form a coherent interpretation of the entire text. The remainder of this section presents a slightly more detailed version of each of these steps.

Textual input is tokenized as a sequence of words. Those words are fed into two processes, a proper name recognizer which would map the string “United States” to the instance UniedStatesOfAmerica, and a lexical lookup process which identifies Cyc lexical concepts that correspond to the input text, for example mapping “death” to Death-TheWord. Those lexical concepts are then fed to another process which identifies Cyc semantic concepts that correspond to the lexical concepts, for example mapping Death-TheWord to the Cyc concept Deceased. These processes leverage assertions in the Cyc knowledge base that map strings to corresponding lexical concepts or named entities.

The outputs of all three of these processes are fed into the DMAP phrasal pattern matcher. Functionally the

Post Sentence 1 Result "An attack occurred in Madrid.	**Post Sentence 2 Result** "The bombing was perpetrated by the ETA."

Pattern:
`<Event> Occur-TheWord In-TheWord <GeographicLocation>`
Assertions:
`(eventOccursAt ?event ?location)`
`(isa ?event AttackOnObject)`
`(isa ?location GeographicLocation)`
	Pattern:
	`<Event> Perpetrate-TheWord By-TheWord <IntelligentAgent>`
	Assertions:
	`(perpetrator ?bombing ?agent)`
	`(isa ?bombing Bombing)`
	`(isa ?agent IntelligentAgent)`

Variables	Values
`?location:`	• `CityOfMadridSpain`
`?event:`	• `TerroristAttack-August-8-2000-Madrid-Spain` • `TerroristAttack-September-8-2003-Madrid` • `TerroristAttack-11-mar-2004-Madrid-Spain`

Variables	Values
`?agent:`	• `BasqueFatherlandAndLiberty`
`?bombing:`	*no values*

Note: the Cyc lexicon defines "ETA" as BasqueFatherlandAndLiberty

Figure 2: Shows the state of DMAP after 2 sentences have been read.

pattern matcher in DMAP is similar to a tabular chart parser (Kay 1996). The pattern matcher has mappings from sequences of strings, lexical, and semantic concepts, to semantic assertions which they can be translated into. Figure 2 shows two such mappings. The first sentence is about where an event occurred. The pattern starts by expecting a reference to an Event, followed by an expectation for the lexical concept Occur-TheWord, etc. The pattern matcher is recursive, and output from one pattern can be used to contribute to a larger pattern. If no patterns are matched DMAP will skip processing that sentence and move on to the next one.

The assertions sets produced by the pattern matcher are fed into an interpretation manager, which is responsible for linking assertions from one sentence to the next, and identifying relevant remindings in the existing memory (KB). Alternate interpretations of ambiguous texts are captured as multiple competing states. DMAP leverages existing memory to help reduce ambiguity by preferring interpretations that reference already existing memory structures.

The critical portion of merging the assertions from one sentence to the next is the alignment of the role variables they contain. This process is combinatorically expensive. DMAP requires that two variables agree with respect to type for them to be merged. Two types agree if one is a generalization of the other, or if DMAP has, in memory, an example of an instance that is a member of both types; again using its memory to determine what parsing alternatives are acceptable and preferred. DMAP will also prefer those alignments that allow it to retrieve more known facts from memory when the bindings from reading are substituted into the assertions. In lieu of that, DMAP will select an alignment that agrees with respect to types, and does not cause the contradiction of existing knowledge when the bindings are substituted. For example, DMAP considers it a contradiction if an assertion is being formed indicating one agent performed an event, when it has knowledge in memory indicating that another actor performed the event. In the example in Figure 2 there are 7 possible alignments of the variables for DMAP to consider, ultimately it will prefer the following alignment.

```
?location :: <no pair>
   ?event :: ?bombing
<no pair> :: ?agent
```

If DMAP can find no remindings in memory for a given interpretation it will produce new Skolem constants as values for any role variables without values retrieved from memory.

Using Episodic Memory

Our current work in language understanding is being done in conjunction with the Learning Reader project. A goal of that research is to be able to use existing knowledge to improve the acquisition and integration of new knowledge, from (simplified) natural language. The Learning Reader is intended to be a complete Machine Reader.

DMAP was selected as the natural language understanding front-end of the Learning Reader because its understanding is driven by memory structures, and it maps immediately and directly to existing knowledge. This provides us with an opportunity to experiment with and evaluate methods for using existing knowledge (both semantic and episodic) to facilitate machine reading.

Three such experiments for leveraging existing knowledge in memory are detailed below.

Performance is evaluated relative to a reference set of assertions. This answer key was built by taking the output of a hand tuned, more exhaustive (very slow) version of DMAP, and then hand evaluating the assertions it produced and removing those which were judged to be incorrect representations of the story content (in a few instances some assertions were added by hand as well). It should be noted that this key is not complete, and that it is an ongoing effort to produce a more complete answer key which represents more (all) of the knowledge present in the test corpus in CycL.

Briefly, it should be mentioned, that the implementations discussed here process text at an average rate of under 2 seconds per sentence, and the median time for a sentence is under 0.5 seconds. Also the simplified text corpus was produced to evaluate the Learning Reader for knowledge acquisition and understanding from text. In that regard the knowledge produced by DMAP allows the Learning Reader to answer 27% more journalist questions (who, what, when…) about the entities and events referenced in the text with a 99.7% accuracy rate, as compared to only using the information contained in Research Cyc alone. (These results are currently in consideration for publication.)

Favoring Existing Knowledge

The hypothesis is that machine reading interpretations that include more knowledge (assertions) that already exists in memory will be better representations of the input text.

This was evaluated by implementing a preference on the Interpretation Manager (see Figure 1) for favoring interpretations after each sentence, which produced more preexisting knowledge. Our implementation already has a preference for producing interpretations which involve phrasal patterns that cover more of the text. When multiple patterns cover the same amount of text, we built one implementation which preferred that which mapped to more preexisting knowledge and one that selected arbitrarily from the ambiguous options.

We did see evidence of an improvement, although it was very small. The system with a preference for existing knowledge reproduced 84% of the assertions in the key while the other system reproduced 83%. Looking at the number of preexisting assertions produced by each version, we see that the one with the preference for preexisting knowledge produced 88 preexisting assertions (not including isa assertions) versus 76 assertions for the other version. Considering that the entire corpus is currently represented in the reference key using 966 assertions, it's not surprising that the difference in performance is as small as 1%.

While this method does show potential for improving results, it is only a tiny one, with respect to our current corpus. The current test corpus was selected to include stories about geography, politics, and military events primarily centered in the Middle East. No effort was made to include or exclude stories which involved information already contained in Cyc. It is clear that much of our corpus represents events that are new to Cyc. A follow-up hypothesis is that, given input text about already known events, more significant improvement might be gained from this heuristic. As a part of the Learning Reader research more extensive corpora are being compiled and one goal is to look at the Learning Reader's performance at re-reading, or reading multiple stories about the same set of events, such as a task analogous to following the development of a current event. It is possible, in this domain, a reuse driven heuristic would perform well.

Shared Instances: Coreference Exemplars

Resolving coreferences within or across sentences is a difficult problem and the source of much research. (Hobbs 1979, McCarthy and Lehnert 1995,)

One common way to leverage memory to assist in coreference resolution is to use type hierarchies (also know as isa hierarchies or sort hierarchies). In such an approach, two references are allowed to be coreferent if one represents a concept or type that is a generalization of the other. For example, "a person" and "the man" could be coreferent since all men are people. However, this problem becomes much more difficult when dealing with sibling categories which do not necessarily imply each other. For example, we found DMAP asking if a reference to a terrorist could be the same as a reference to a married person. One collection is not the generalization of the other, and it's certainly not the case that all terrorists are married or visa versa. A human reader, however, can easily recognize that a terrorist could be a married person.

Existing semantic knowledge could be leveraged to reason about whether or not these type are mutually exclusive, or if a member of one is allowed to be a member of the other, etc. It was our hypothesis that this problem could be addressed more simply by using episodic memory, and taking a case-based reasoning approach. We proposed the algorithm that two references could be coreferent if one was a generalization of the other, or if an instance belonging to both collections existed in memory. This makes the question one of retrieval "do I know of an instance of X that's also an instance of Y?", as opposed to one of reasoning, "can I prove that instances of X can/cannot be instances of Y?".

We saw a 2% improvement 85% of the key returned by the implementation which used generalization and shared instance information to allow coreferences, verses 83% produced by a version which only used generalization information to permit coreferences. Looking more closely at the numbers only 7% of the sentences induced DMAP to ask if a coreference was possible based on shared instance information, and only 14% of the reference alignments it produced relied on a shared instance to complete the alignment. We refer to an alignment as the resolution of references from one sentence to those references contained in all of the previous sentences in that story. (One alignment operation could involve evaluating several opportunities for coreference.) For this particular reading of our corpus there were 1,720 alignment operations performed, 1,475 of which were possible using only generalization information.

The current test corpus was compiled prior to these evaluations, and so there was not any particular focus on producing input texts which would evaluate DMAP's performance resolving various forms of coreference. Also our corpus is written in simplified English. Vocabulary is not restricted, but a preference was made for simplified grammatical structure, as can be seen in the example story in the second section of this paper. Although there were no restrictions on those producing the simplified English regarding reference, it turned out to be easier for the writers to repeat the same reference over and over again, "An attack…", "the attack…", etc. instead of varying it between multiple references, such as "bombing", "terrorist attack", "explosion", etc. As a result of which our current corpus does not contain many opportunities to evaluate DMAP's performance regarding more complicated coreferences.

Predicate Co-occurrence

The third heuristic we evaluated was inspired by the first (a preference for preexisting knowledge). The hypothesis is that if specific episodic information about the current story cannot be identified in memory to retrieve, generalized episodic information might still be of use to resolve ambiguity. The preference here is for interpretations of a given sentence that use predicates that frequently co-occur with the predicates used to represent the interpretation of the preceding portions of the same text.

For example, when reading about the geography of a country, the sentence X "has a border with" Y, or variations thereof, may be encountered. In a particularly bad parse it may be possible to get fragments which translate the "has" into a statement of ownership, while the "border" part produces a competing interpretation about the borders of two regions. Given the context of the story, and the predicates used to represent it, it was hypothesized that it was more likely for those predicates to co-occur with assertions regarding borders than possession, etc.

Co-occurrence was computed across the entire Cyc KB, two predicates were considered to co-occur every time there was an instance of each predicate that shared the same argument. This was kind of a naive construction of a "poor man's" frame structure from the logical assertions.

This information was put to use in two places, one was to order competing interpretations when they both produced the same amount of preexisting data. The second place was to restrict ambiguity when there were numerous phrasal patterns matching the same piece of text. The previous algorithm would branch and create competing states for each option if the branching was low, but skip the sentence if the branching was too high. In the experiment, the new algorithm would select only the top n states, based on predicate co-occurrence and continue with those states, instead of skipping the sentence.

Allowing it to pick the top 4 states based on co-occurrence produced a 2% increase in answer key reproduction from 85% to 87%. Raising it to 8 states had no benefit over using only 4, and really just introduced more ambiguity for DMAP to spend time sorting out.

Future Work

On the more pragmatic end, future work will involve moving to the latest version of ResearchCyc which has over twice as many assertions as the version we are currently using. The new version contains more linguistic information as well which can be mined to produce more phrasal patterns for DMAP. Possibly the additional episodic knowledge will affect the algorithms discussed in this paper.

It is ongoing work of the Learning Reader to continue to augment and extent our test corpus. Producing or identifying corpora designed for evaluating particular natural language tasks, such as coreference, will also be important, in order to evaluate the bounds of DMAP.

On the other end of the spectrum, future work will include identify areas where the performance of DMAP can be significantly improved, for example, mining script-like information out of Cyc, to produce patterns for higher level knowledge structures. Having such information would allow DMAP to recognize semantic structures that span sentences, reducing the amount of coreference resolution required, and turning more of the process into recognition instead of reasoning or evaluation.

Summary

Direct Memory Access Parsing (DMAP) is a machine reading approach that leverages existing knowledge and performs integration in the early stages of parsing natural language text. DMAP's understanding is driven by memory structures, and it maps immediately and directly to existing knowledge. This provides an opportunity to experiment with and evaluate methods for using existing knowledge, both semantic and episodic, to facilitate machine reading. Three such experiments for leveraging existing episodic knowledge in memory were presented.

The techniques presented showed a 1-2% performance increases each on precision, for our test corpus. We believe that these ideas still warrant further investigation, and probably experimentation with corpora more suited to evaluate under what conditions their use is warranted and effective.

Machine reading is fundamentally about integrating new knowledge gleaned from reading with existing knowledge. We have begun to explore how mapping to and using existing knowledge early in the language understanding process can be leveraged to facilitate this integration.

Acknowledgements

The authors would like to acknowledge the other members of the Learning Reader research group for their discussion and help which contributed to this paper: Ken Forbus, Larry Birnbaum, Abhishek Sharma, and Leo Ureel. This project was funded through DARPA, grant HR0011-04-1-0051. We would like to thank Cycorp for access to and support with using Research Cyc. The Qualitative Reasoning Group (QRG) at Northwestern has also been essential to this work through ongoing support with using their Fire reasoning engine (Forbus 1993).

References

Calder, J., Klein, E., and Zeevat, H. 1988. *Unification Categorial Grammar: a concise, extendable grammar for natural language processing.* Proceedings of the 12th Conference on Computational Linguistics - Volume 1 (Budapest, Hungry, August 22 - 27, 1988). D. Vargha, Ed. International Conference On Computational Linguistics. Association for Computational Linguistics, Morristown, NJ, 83-86.

Clark, P., Harrison, P., and Thompson, J. 2003. *A knowledge-driven approach to text meaning processing.* In Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning - Volume 9 Human Language Technology Conference. Association for Computational Linguistics, Morristown, NJ, 1-6.

Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. 2004. *Web-scale information extraction in knowitall: (preliminary results).* In Proceedings of the 13th international Conference on World Wide Web (New York, NY, USA, May 17 - 20, 2004). WWW '04. ACM Press, New York, NY, 100-110.

Forbus, Kenneth D., de Kleer, Johan. 1993 *Building problem solvers*, MIT Press, Cambridge, MA, 1993

Hobbs, J. 1979. *Coherence and coreference.* Cognitive Science 3:67--90.

Kay, M. 1996 *Chart generation*. In Proceedings of the 34th Annual Meeting on Association For Computational Linguistics (Santa Cruz, California, June 24 - 27, 1996). Annual Meeting of the ACL. Association for Computational Linguistics, Morristown, NJ, 200-204.

Lenat, D. B. 1995 *CYC: A Large-Scale Investment in Knowledge Infrastructure.* Communications of the ACM 38 (11): 33–38.

Manning, C. and Schutze, H. 1999 *Foundations of Statistical Natural Language Processing.* MIT Press Cambridge, Massachusetts, 1999.

Martin, C. E. 1990 *Direct Memory Access Parsing.* PhD thesis, Yale University. 1990.

McCarthy, J. and W. Lehnert. 1995. *Using decision trees for coreference resolution.* In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, pages 1050-1055.

Pease, A.; Murray, W., 2003 *An English to logic translator for ontology-based knowledge representation languages*, Proceedings of International Conference on Natural Language Processing and Knowledge Engineering, 2003, vol., no.pp. 777- 783, 26-29 Oct. 2003

Riesbeck, C. K. 1986 *From Conceptual Analyzer to Direct Memory Access Parsing: An Overview.* Advances in Cognitive Science. pages 236-258 N. E. Sharkey (ed.), Prentice Hall. 1986.

Riesbeck, C. and Martin, C., 1985 *Direct Memory Access Parsing*, Yale University Report 354, 1985.