Improving the Quality of Text Understanding by Delaying Ambiguity Resolution

Doo Soon Kim
Dept. of Computer Science
University of Texas
onue5@cs.utexas.edu

Ken Barker
Dept. of Computer Science
University of Texas
kbarker@cs.utexas.edu

Bruce Porter
Dept. of Computer Science
University of Texas
porter@cs.utexas.edu

Abstract

Text Understanding systems often commit to a *single best* interpretation of a sentence before analyzing subsequent text. This interpretation is chosen by resolving ambiguous alternatives to the one with the highest confidence, given the context available at the time of commitment. Subsequent text, however, may contain information that changes the confidence of alternatives. This may especially be the case with multiple redundant texts on the same topic. Ideally, systems would delay choosing among ambiguous alternatives until more text has been read.

One solution is to maintain multiple candidate interpretations of each sentence until the system acquires disambiguating evidence. Unfortunately, the number of alternatives explodes quickly. In this paper, we propose a *packed graphical (PG) representation* that can efficiently represent a large number of alternative interpretations along with dependencies among them. We also present an algorithm for combining multiple PG representations to help resolve ambiguity and prune alternatives when the time comes to commit to a single interpretation.

Our controlled experiments show that by delaying ambiguity resolution until multiple texts have been read, our prototype's accuracy is higher than when committing to interpretations sentence-by-sentence.

1 Introduction

A typical text understanding system confronts ambiguity while parsing, mapping words to concepts and formal relations, resolving co-references, and integrating knowledge derived from separate sentences or texts. The system discards many candidate interpretations to avoid combinatorial explosion. Commonly, after reading each sentence, a system will commit to its top ranked interpretation of the sentence before reading the next.

If a text understanding system could postpone committing to an interpretation without being swamped by a combinatorial explosion of alternatives, its accuracy would almost surely improve. This intuition follows from the observation that text is redundant in at least two ways. First, within a single coherent text (about the same entities and events), each sentence informs the interpretation of its neighbors. Second, within a corpus of texts on the same topic, the same information is expressed in different surface forms, ambiguous in different ways. Related fields, such as Information Extraction, exploit textual redundancy to good effect, and perhaps text understanding can as well.

One approach is for the text understanding system to maintain multiple complete candidate interpretations. After reading each sentence, for example, the system would retain a beam of the n-best interpretations of the sentence. While this approach avoids a combinatorial explosion (for reasonable values of n), several problems remain. First, because the beam width is limited, the system may still discard correct interpretations before benefiting from the extra context from related text. Second, enumeration of the candidate interpreta-

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 581–589,
Beijing, August 2010

tions does not represent the dependencies among them. For example, there may be multiple candidate word senses and semantic roles for a given sentence, but sense alternatives might be dependent on role selection (and vice-versa). The set of reasonable interpretations may be a subset of all combinations. Finally, maintaining distinct interpretations does not contribute to addressing the problem of combining evidence to narrow down alternatives and ultimately select a single best interpretation of a text.

This paper addresses these three problems. We propose an approach that postpones committing to an interpretation of a text by representing ambiguities and the dependencies among them. There may still be combinatorial growth in the set of alternative interpretations, but they are represented only intensionally, using a packed representation, which maintains alternatives while avoiding enumerating them. We also propose an algorithm for updating and pruning the packed representation as more sentences and texts are read.

We evaluate our approach by comparing two reading systems: a baseline system that commits to its best interpretation after each sentence, and our prototype system that uses a packed representation to maintain all interpretations until further reading enables it to prune. For this initial proof of concept, we use a small corpus of redundant texts. The results indicate that our approach improves the quality of text interpretation by preventing aggressive pruning while avoiding combinatorial explosion.

In the following sections, we first describe our target semantic representation of the interpretation of sentences. We then present the details of our *packed graphical representation (PG representation)* and our algorithm to resolve ambiguities in the PG representations as disambiguating evidence from subsequent text accrues. We describe the architecture of a prototype that produces PG representations for text and implements the disambiguating algorithm. Finally, we present the results from controlled experiments designed to compare the accuracy of the prototype to a baseline system that prunes more aggressively.

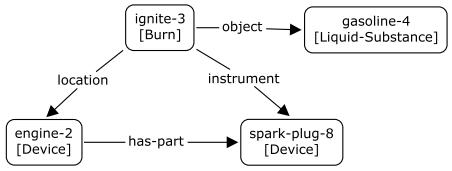

Figure 1: The target semantic graph representation for S1

2 Target semantic representation

Our target representation is a semantic graph in which nodes are words and the ontological types to which they map. Edges are semantic relations corresponding either to function words or syntactic relations in the sentence's parse.

Fig. 1 shows the target semantic representation for the following simple sentence:

S1: *An engine ignites gasoline with its spark plug.*

3 PG representation

Alternative semantic interpretations for a sentence can be captured with a single PG representation with ambiguities represented as local alternatives. Because candidate representations are often structurally similar, a PG representation can significantly compress the representation of alternatives.

Fig. 2 shows the PG representation of alternate interpretations of S1 (PG1). The different types of ambiguity captured by the PG representation are as follows.

3.1 Word-Type ambiguity

In PG1, the node engine-2a corresponds to the word "engine" in S1. Its annotation [LIVING-ENTITY .3 | DEVICE .7] captures the mapping to either LIVING-ENTITY (probability 0.3) or DEVICE (probability 0.7). The PG representation does not presume a particular uncer-

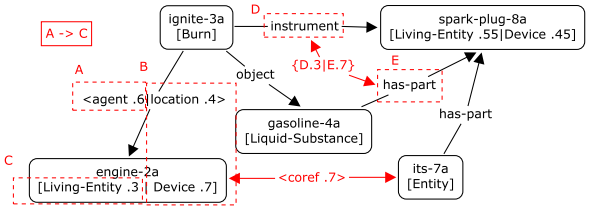

Figure 2: The PG representation for S1 (PG1)

tainty formalism. Any formalism, (Dempster-Shafer theory (Pearl, 1988), Markov Logic Networks (Richardson and Domingos, 2006), etc.) could be used.

3.2 Semantic Relation ambiguity

In PG1, the edge label <agent .6 | location .4> from ignite-3a to engine-2a says that the engine is either *agent* or *location* of the ignition.

3.3 Structural ambiguity

In PG1, edges D and E are alternatives corresponding to the different prepositional phrase attachments for "with its spark plug" (to ignite-3a or gasoline-4a). The annotation {D .3 | E .7} says that the choices are mutually exclusive with probabilities of 0.3 and 0.7.

3.4 Co-reference ambiguity

Co-reference of nodes in a PG representation is captured using a "co-reference" edge. In PG1, the edge labeled <coref .7> represents the probability that engine-2a and its-7a are co-referent.

In addition to storing ambiguities explicitly, the PG representation also captures dependencies among alternatives.

3.5 Simple dependency

The existence of one element in the graph depends on the existence of another element. If subsequent evidence suggests that an element is incorrect, its dependents should be pruned. For example, the dependency A $\rightarrow$ C, means that if LIVING-ENTITY is ultimately rejected as the type for engine-2a, the agent relation should be pruned.

3.6 Mutual dependency

Elements of a mutual dependency set are mutually confirming. Evidence confirming or rejecting an element also confirms or rejects other elements in the set. In the example, the box labeled B says that (engine-2a type DEVICE) and (ignite-3a location engine-2a) should both be confirmed or pruned when either of them is confirmed or pruned.

Formally, the PG representation is a structure consisting of (a) *semantic triples* – e.g., (ignite-3a type BURN), (b) *macros* – e.g., the symbol A refers to (ignite-3a agent engine-2a), and (c) *constraints* – e.g., A depends on C.

4 Combining PG representations

Maintaining ambiguity within a PG representation allows us to delay commitment to an interpretation until disambiguating evidence appears. For any text fragment that results in a PG representation (PGa) containing ambiguity, there may exist other text fragments that are partly redundant, but result in a less ambiguous (or differently ambiguous) representation (PGb). PGb can be used to adjust confidences in PGa. Enough such evidence allows us to prune unlikely interpretations, ultimately disambiguating the original representation.

For example, sentence S3 does not have sufficient context to disambiguate between the MOTOR sense of "engine" and the VEHICLE sense (as in *locomotive*).

S3: *General Electric announced plans this week for their much anticipated new engine.*

The PG3 representation for S3 (PG3) would maintain the ambiguous representation (with confidences for each sense based on prior probabilities, for example). On subsequently encountering sentence S4, a Lesk-based word sense disambiguation module (as in our prototype) would produce a PG4 with a strong preference for the locomotive sense of "engine", given the more specific context of S4.

S4: *The announcement comes to the relief of many in the railway industry looking to replace the engines in their aging locomotive fleets.*

To use PG4 to help disambiguate PG3, we need to align PG3 and PG4 semantically and merge their conflict sets. (In the simple example, the conflict sets for the word "engine" might be [MOTOR .5 | VEHICLE .5] in PG3 and [MOTOR .2 | VEHICLE .8] in PG4).

Algorithm 1 describes how two PG representations can be combined to help resolve their ambiguities. The algorithm identifies their isomorphic subgraphs (redundant portions of the interpretations) and uses the information to disambiguate their ambiguities. For illustration, we will step through Algorithm 1, merging PG1 (Fig. 2) with

Algorithm 1 Disambiguating PG representations

Input : PG1, PG2

Output: new PG representation

1. *Identify semantically aligned parts between PG1 and PG2.* Use graph matching to identify alignments (redundant portions) between PG1 and PG2: align nodes with the same base word or with taxonomically related types; from the node alignments, align identical types as type alignments; align relations if the relations are the same and their head and tail nodes have been aligned.

2. *Use alignments to disambiguate PG1 and PG2.* With the available information (the confidence scores and the constraints in PG1 and PG2 and the alignments between them), use joint inference to calculate the confidence score of each candidate interpretation. If the confidence score of one interpretation becomes much higher than competing ones, the interpretation is chosen while the others are discarded.

3. *Combine the disambiguated PG1 and PG2 into one PG representation using the alignments identified in the first step.*

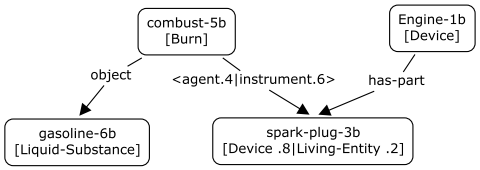

Figure 3: PG representation for S2, *"The engine's spark plug combusts gasoline."*

PG2 (Fig. 3).

1. The graph matcher identifies alignments between PG1 and PG2. Type alignments include (engine-2a[DEVICE], Engine-1b[DEVICE]), (spark-plug-8a[LIVING-ENTITY], spark-plug-3b[LIVING-ENTITY]). Relation alignments include ((combust-5b instrument spark-plug-3b), (ignite-3 instrument spark-plug-8)), ((ignite-3a instrument spark-plug-8a) (combust-5b instrument spark-plug-3b)).

2. In this example, when two interpretations are aligned, we simply add their confidence scores. (We are currently incorporating *Alchemy* (Richardson and Domingos, 2006) in the prototype system to do the joint inference). For example, aligning engine-2a with Engine-1b results in a score of 1.7 for DEVICE (1 + .7). The confidence score of LIVING-ENTITY in engine-2a is unchanged at .3. Since the resulting score for DEVICE is much higher than [1] the score for LIVING-ENTITY, LIVING-ENTITY is discarded. Deleting LIVING-ENTITY causes deletion of the *agent* edge between ignite-3a and engine-2a due to the dependency constraint A $\rightarrow$ C.

3. The disambiguated PG1 and PG2 are merged into a single PG representation (PG1+2) based on the alignments. Any remaining ambiguity persists in PG1+2, possibly to be resolved with another sentence.

5 Prototype system

5.1 Parser

Our prototype system uses the Stanford Parser (Klein and Manning, 2003). To capture structural ambiguity for our experiments, we manually edited the parser output by adding corrections as alternatives wherever the parse tree was incorrect. This gave a syntactic PG representation with both incorrect and correct alternatives. We gave the original, incorrect alternatives high confidence scores and the added, correct alternatives low scores, simulating a parser pruning correct interpretations in favor of incorrect ones with higher confidence scores. The syntactic PG for S1 is shown in Fig. 4. We have recently designed a modification to the Stanford Parser to make it produce syntactic PG representations natively, based on the complete chart built during parsing.

5.2 Semantic Interpreter

The semantic interpreter assigns types to nodes in the syntactic PG representation and semantic relations to the edges.

Type ambiguity. Types and confidence scores are assigned to words using SenseRelate (Patwardhan et al., 2005), WSD software based on the

[1] In our prototype, we set the pruning threshold at $\frac{1}{3}\times$the score of the top-scored interpretation.

Lesk Algorithm (Lesk, 1986). Assigned senses are then mapped to our *Component Library* ontology (Barker et al., 2001) using its built-in WordNet mappings.

Relational ambiguity. Semantic relations are assigned to the dependency relations in the syntactic PG representation according to semantic interpretation rules. Most rules consider the head and tail types as well as the dependency relation, but do not produce confidence scores. Our prototype scores candidates equally. We plan to incorporate a more sophisticated scoring method such as (Punyakanok et al., 2005).

Structural ambiguity. Parse ambiguities (such as PA vs. PB in Fig. 4) are converted directly to structural ambiguity representations (D vs. E in Fig. 2) in the semantic PG representation.

Simple Dependency. A dependency is installed between a type t for word w and a semantic relation r when (1) r is produced by a rule based on t and (2) r is dependent on no other candidate type for w. In Fig. 2, a dependency relation is installed from A to C, because (1) LIVING-ENTITY in engine-2a was used in the rule assigning *agent* between ignite-3a and engine-2a and (2) the assignment of *agent* is not dependent on DEVICE, the other candidate type of engine-2a.

Mutual dependency. If multiple interpretations depend on one another, a mutual dependency set is created to include them.

5.3 PG Merger

The PG Merger implements Algorithm 1 to combine PG representations. The PG representation

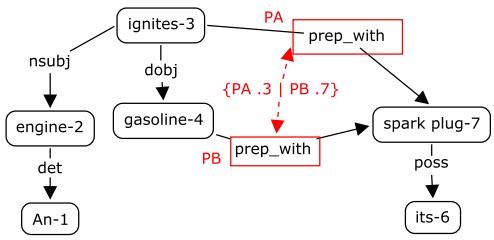

Figure 4: Syntactic PG representation for S1, capturing the PP-attachment ambiguity of "with its spark plug".

Original Text Hearts pump blood through the body. Blood carries oxygen to organs throughout the body. Blood leaves the heart, then goes to the lungs where it is oxygenated. The oxygen given to the blood by the lungs is then burned by organs throughout the body. Eventually the blood returns to the heart, depleted of oxygen.
Paraphrase The heart begins to pump blood into the body. The blood first travels to the lungs, where it picks up oxygen. The blood will then be deposited into the organs, which burn the oxygen. The blood will then return to the heart, where it will be lacking oxygen, and start over again.

Figure 5: The original text and a paraphrase

for each sentence is merged with the cumulative PG from previous sentences. The global PG representation integrates sentence-level PG representations to the extent that they align semantically. In the worst case (completely unrelated sentences), the global PG representation would simply be the union of individual PG representations. The extent to which the global PG is more coherent reflects redundancy and semantic overlap in the sentences.

6 Experiment 1

We first wanted to evaluate our hypothesis that Algorithm 1 can improve interpretation accuracy over multiple redundant texts. We manually generated ten redundant texts by having volunteers rewrite a short, tutorial text, using Amazon Turk (http://mturk.com) [2] The volunteers had no knowledge of the purpose of the task, and were asked to rewrite the text using "different" language. Fig. 5 shows the original text and one volunteer's rewrite. The total number of sentences over the ten texts was 37. Average sentence length was 14.5 words.

6.1 Evaluation Procedure

We ran two systems over the ten texts. The baseline system commits to the highest scoring consistent interpretation after each sentence. The prototype system produces an ambiguity-preserving

[2] We ultimately envision a system whose task is to develop a model of a particular topic by interpreting multiple texts. Such a system might be given a cluster of documents or use its own information retrieval to find similar documents given a tutorial text.

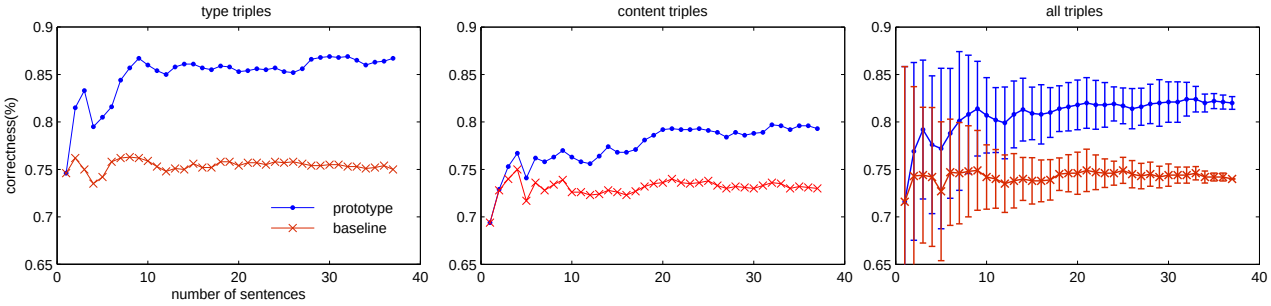

Figure 6: Correctness scores for the prototype vs. baseline system on (a) type triples (word sense assignment), (b) content triples (semantic relations) and (c) all triples (with standard deviation).

PG representation. For each sentence, the prototype's PG Merger merges the PG of the sentence with the merged PG of the previous sentences. After N sentences (varying N from 1..37), the system is forced to commit to the highest scoring consistent interpretation in the merged PG. For N=1 (commit after the first sentence), both the baseline and prototype produce the same result. For N=2, the baseline produces the union of the highest scoring interpretations for each of the first two sentences. The prototype produces a merged PG for the first two sentences and then prunes to the highest scoring alternatives.

At each value of N, we measured the correctness of the interpretations (the percentage of correct semantic triples) for each system by comparing the committed triples against human-generated gold standard triples.

We repeated the experiment ten times with different random orderings of the 37 sentences, averaging the results.

6.2 Evaluation result

Fig. 6 shows that both type assignment and semantic relation assignment by the prototype improve as the system reads more sentences. This result confirms our hypothesis that delaying commitment to an interpretation resolves ambiguities better by avoiding overly aggressive pruning.

To determine an upper bound of correctness for the prototype, we inspected the PG representations to see how many alternative sets contained the correct interpretation even if not the highest scoring alternative. This number is different from the correctness score in Fig. 6, which is the percentage of gold standard triples that are the highest scoring alternatives in the merged PG.

	baseline	prototype
nodes w/ the correct type	76	**91**
edges w/ the correct relation	74	**88**

Table 1: Percentage of nodes and edges containing the correct types and semantic relations in the baseline and the prototype for all 37 sentences.

Table. 1 shows that 91% of the nodes in the PG contain the correct type (though not necessarily the highest scoring). 88% of the edges contain the correct semantic relations among the alternatives. In contrast, the baseline has pruned away 24% of the correct types and 26% of the correct semantic relations.

7 Experiment 2

Our second experiment aims to evaluate the claim that the prototype can efficiently manage a large number of alternative interpretations. The top line in Fig. 7 shows the number of triples in the PG representations input to the prototype. This is the total number of triples (including ambiguous alternatives) in the PG for each sentence prior to invoking Algorithm 1. The middle line is the number of triples remaining after merging and pruning by Algorithm 1. The bottom line is the number of triples after pruning all but the highest scoring alternatives (the baseline system). The results show that Algorithm 1 achieves significant compression over unmerged PG representations. The resulting size of the merged PG representations more closely tracks the size of the aggressively pruned

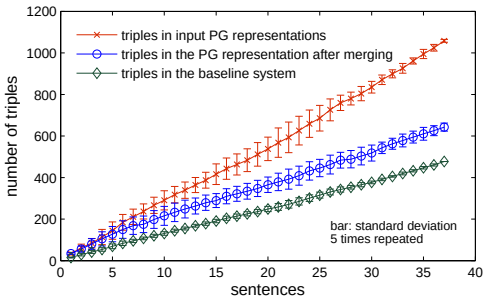

Figure 7: Total number of triples in individual sentence PG representations (top); total number of triples in the PG representation after merging in the prototype system (middle); total number of triples after pruning to the highest scoring alternative (bottom).

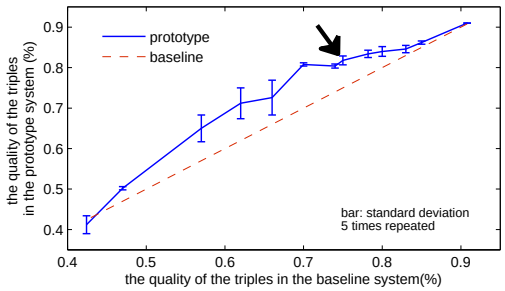

Figure 8: Sensitivity of the prototype and baseline systems to the quality of the NL system output. The quality of input triples is perturbed affecting performance accuracy of the two systems. For example, when the quality of input triples is such that the baseline system performs at 70% accuracy, the prototype system performs at 80%. The arrow indicates unperturbed language interpreter performance.

representations.

8 Experiment 3

Finally, we wanted to measure the sensitivity of our approach to the quality of the natural language interpretation. In this experiment, we artificially varied the confidence scores for the correct interpretations in the PG representations input to the prototype and baseline systems by a fixed percentage. For example, consider a node heart-1 with multiple candidate types, including the correct sense for its context: INTERNAL-ORGAN with confidence 0.8. We reran Experiment 1 varying the confidence in INTERNAL-ORGAN in increments of +/-10%, while scaling the confidences in the incorrect types equally. As the confidence in correct interpretations is increased, all correct interpretations become the highest scoring, so aggressive pruning is justified and the baseline performance approaches the prototype performance. As the confidences in correct interpretations are decreased, they are more likely to be pruned by both systems.

Fig. 8 shows that Algorithm 1 is able to recover at least some correct interpretations even when their original scores (relative to incorrect alternatives) is quite low.

9 Discussion and Future Work

Our controlled experiments suggest that it is both desirable and feasible to delay ambiguity resolution beyond sentence and text boundaries. Improvements in the correctness of semantic interpretation of sentences is possible without an explosion in size when maintaining multiple interpretations.

Nevertheless, these experiments are proofs of concept. The results confirm that it is worthwhile to subject our prototype to a more real-world, practical application. To do so, we need to address several issues.

First, we manually simulated structural (parse) ambiguities. We will complete modifications to the Stanford Parser to produce PG representations natively. This change will result in a significant increase in the number of alternatives stored in the PG representation over the current prototype. Our initial investigations suggest that there is still enough structural overlap among the candidate parse trees to allow the PG representation to control explosion, but this is an empirical question that will need to be confirmed.

We are modifying our semantic interpreter to admit induced semantic interpretation rules which will allow us to train the system in new domains.

The current prototype uses a naive heuristic for identifying co-reference candidates. We are investigating the use of off-the-shelf co-reference systems.

Finally, we are incorporating the Alchemy (Richardson and Domingos, 2006)

probabilistic inference engine to calculate the probability that a candidate interpretation is correct given the PG constraints and alignments, in order to inform confirmation or pruning of interpretations.

Once these updates are complete, we will perform more wide-scale evaluations. We will investigate the automatic construction of a test corpus using text clustering to find redundant texts, and we will conduct experiments in multiple domains.

10 Related Work

Succinctly representing multiple interpretations has been explored by several researchers. The packed representation (Maxwell III and Kaplan, 1981; Crouch, 2005) uses logical formulae to denote alternative interpretations and treats the disambiguation task as the propositional satisfiability problem. Core Language Engine (Alshawi, 1992) introduces two types of packing mechanism. First, a quasi logical form allows the underspecification of several types of information, such as anaphoric references, ellipsis and semantic relations (Alshawi and Crouch, 1992). Second, a packed quasi logical form (Alshawi, 1992) compactly represents the derivations of alternative quasi logical forms. In contrast, the PG representation is (1) based on a graphical representation, (2) explicitly represents constraints and (3) includes confidence scores.

These representations and the PG representation have one feature in common: they represent a set of complete alternative interpretations of a text. Another class of compact representations, called "underspecification", has been studied as a formal representation of ambiguous sentences. These representations include Hole Semantics (Bos, 2004), Underspecified Discourse Representation Semantics (Reyle, 1995), Minimal Recursion Semantics (Copestake et al., 2005) and Dominance Constraints (Egg et al., 2001). These representations, rather than packing fully-represented candidate interpretations, specify fragments of interpretations which are unambiguously interpreted, along with constraints on their combination (corresponding to different interpretations). They generally focus on specific ambiguities such as scope ambiguity (Bos, 2004) (Egg et al., 2001) (Copestake et al., 2005) or discourse relations (Schilder, 1998) (Regneri et al., 2008).

Disambiguating compact representations has received relatively less attention. (Riezler et al., 2002; Geman and Johnson, 2002) use a packed representation to train parsers on a corpus and uses the learned statistics to disambiguate packed representations. (Clark and Harrison, 2010) uses paraphrase databases and a hand-built knowledge base to resolve underspecified representations.

Different architectures have been proposed to improve the pipeline architecture. (Sutton and McCallum, 2005; Wellner et al., 2004) maintain a beam of n best interpretations in the pipeline architecture. Their pipeline, however, consists of only two components. (Finkel et al., 2006) uses sampling over the distribution of alternative interpretations at each stage of the pipeline and then passes the sampled data to the next component. The packed representation (Crouch, 2005) and CLE (Alshawi, 1992) use packed representation in the pipeline, though both, at some stages, unpack them and re-pack the processed result. (Crouch and King, 2006) later proposes a new method that does not require unpacking and then repacking.

11 Conclusion

We have begun to address the challenge of efficiently managing multiple alternative interpretations of text. We have presented (1) a *packed graphical representation* that succinctly represents multiple alternative interpretations as well as the constraints among them, and (2) an algorithm for combining multiple PG representations to reinforce correct interpretations and discount implausible interpretations. Controlled experiments show that it is possible to improve the correctness of semantic interpretations of text by delaying disambiguation, without incurring the cost of an exponentially expanding representation.

12 Acknowledgement

Support for this research was provided in part by Air Force Contract FA8750-09-C-0172 under the DARPA Machine Reading Program

References

Alshawi, Hiyan and Richard S. Crouch. 1992. Monotonic semantic interpretation. In *ACL*, pages 32–39.

Alshawi, Hiyan, editor. 1992. *The Core Language Engine*. MIT Press, Cambridge, Massachusetts.

Barker, Ken, Bruce Porter, and Peter Clark. 2001. A library of generic concepts for composing knowledge bases. In *Proceedings of the international conference on Knowledge capture*, pages 14–21.

Bos, Johan. 2004. Computational semantics in discourse: Underspecification, resolution, and inference. *Journal of Logic, Language, and Information*, 13(2):139–157.

Clark, Peter and Phil Harrison. 2010. Exploiting paraphrases and deferred sense commitment to interpret questions more reliably. In *To appear in Proceedings of CoLing 2010*.

Copestake, Ann, Dan Flickinger, Carl Pollard, and Ivan Sag. 2005. Minimal recursion semantics: an introduction. *Research on Language and Computation*, 3:281–332.

Crouch, Richard S. and Tracy Holloway King. 2006. Semantics via f-struecture rewriting. In *Proceedings of LFG06 Conference*.

Crouch, Dick. 2005. Packed rewriting for mapping semantics to kr. In *In Proceedings Sixth International Workshop on Computational Semantics*.

Egg, Markus, Alexander Koller, and Joachim Niehren. 2001. The constraint language for lambda structures. *Journal of Logic, Language, and Information Vol 10 (4), 2001, pp.457-485*, 10:457–485.

Finkel, Jenny Rose, Christopher D. Manning, and Andrew Y. Ng. 2006. Solving the problem of cascading errors: approximate bayesian inference for linguistic annotation pipelines. In *EMNLP*, pages 618–626, Morristown, NJ, USA.

Geman, Stuart and Mark Johnson. 2002. Dynamic programming for parsing and estimation of stochastic unification-based grammars. In *ACL*, pages 279–286.

Klein, Dan and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *ACL*, pages 423–430, Morristown, NJ, USA.

Lesk, Michael. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26, New York, NY, USA.

Maxwell III, John T. and Ronald M. Kaplan. 1981. A method for disjunctive constraint satisfaction. In Tomita, Masaru, editor, *Current Issues in Parsing Technology*, pages 173–190. Kluwer Academic Publishers, Dordrecht.

Patwardhan, Siddharth, Satanjeev Banerjee, and Ted Pedersen. 2005. Senserelate:: Targetword-A generalized framework for word sense disambiguation. In *ACL*.

Pearl, Judea. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan-Kaufmann.

Punyakanok, Vasin, Dan Roth, and Wen tau Yih. 2005. The necessity of syntactic parsing for semantic role labeling. In Kaelbling, Leslie Pack and Alessandro Saffiotti, editors, *IJCAI*, pages 1117–1123. Professional Book Center.

Regneri, Michaela, Markus Egg, and Alexander Koller. 2008. Efficient processing of underspecified discourse representations. In *HLT*, pages 245–248, Morristown, NJ, USA.

Reyle, Uwe. 1995. Underspecified discourse representation structures and their logic. *Logic Journal of the IGPL*, 3(2-3):473–488.

Richardson, Matthew and Pedro Domingos. 2006. *Markov logic networks*. Kluwer Academic Publishers.

Riezler, Stefan, Tracy H. King, Ronald M. Kaplan, Richard S. Crouch, John T. Maxwell III, and Mark Johnson. 2002. Parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques. In *ACL*, pages 271–278.

Schilder, Frank. 1998. An underspecified segmented discourse representation theory (USDRT). In *COLING-ACL*, pages 1188–1192.

Sutton, Charles and Andrew McCallum. 2005. Joint parsing and semantic role labeling. In *CONLL*, pages 225–228, Morristown, NJ, USA.

Wellner, Ben, Andrew McCallum, Fuchun Peng, and Michael Hay. 2004. An integrated, conditional model of information extraction and coreference with application to citation matching. In *UAI*, pages 593–601, Arlington, Virginia, United States.