Understanding Mental States in Natural Language

Wei Chen
Language Technologies Institute, Carnegie Mellon University
Pittsburgh, PA 15213, USA
weichen@cs.cmu.edu

Abstract

Understanding mental states in narratives is an important aspect of human language comprehension. By "mental states" we refer to beliefs, states of knowledge, points of view, and suppositions, all of which may change over time. In this paper, we propose an approach for automatically extracting and understanding multiple mental states in stories. Our model consists of two parts: (1) a parser that takes an English sentence and translates it to some semantic operations; (2) a mental-state inference engine that reads in the semantic operations and produces a situation model that represents the meaning of the sentence. We present the performance of the system on a corpus of children stories containing both fictional and non-fictional texts.

1 Introduction

Natural language involves statements that carry distinct world-views. By world-views we refer to states of belief, supposition, intention, advice, perceived reality, as well as situations expressed by tenses in natural language such as past, present and future. In this paper, we call these world-views "mental states". Mental states are common phenomena. They span various domains of natural language. Sentence (1a) and sentence (1b) are examples drawn from two different domains: online news articles and fairy tales.

(1) a. The police believe the thieves were trying to steal a solar panel from Sarah's tin roof.[1]

[1] Excerpt from BBC online news: http://news.bbc.co.uk/1/hi/world/africa/7609872.stm

b. She (little red-cap) was surprised to find the cottage-door standing open.[2]

Both of these two sentences involve multiple mental states, which may be nested in one another. Sentence (1a) involves the police's belief and the intention of the thieves in the police's belief; sentence (1b) contains little red-cap's old belief and her updated belief, both of which may be different from the reality. Since the information in mental states is rich and often important, we need some processing technique to extract that information and understand it.

There are two problems in extracting and understanding mental states. First, extracting mental states from text requires recognizing linguistic patterns of mental states. Related problems such as subjectivity recognition have been studied intensively in the natural language processing community. The problem covers various aspects of the "private state frame" [10], including recognizing private states, the sources of private states, the intensity and types of attitudes, among many others. Second, mental states extracted from natural language need to be encoded in some representation form and can be retrieved for further inference. There exists many systems that implements nested evolving beliefs (e.g. [1]), but they generally lacked the ability to draw inference directly from natural language.

In this paper, we propose an approach to extract and represent mental states based on different mental contexts such as one's belief, intention and supposition. Our approach utilizes the *mental spaces theory* [4] in cognitive linguistics. Our goal for the mental state extraction step is to identify *space-builders* that establish new mental contexts (or spaces) or the ones that refer to an existing mental context. The main body of our space-builders consists of agent and psych-term pairs such as "the police believe" (refers to the police's belief context) and "little red-cap was surprised" (refers to multiple belief contexts of little red-cap)[3]. Different objects and propositions are then bundled in these mental contexts. In the mental state understanding step, the mental contexts are instantiated and maintained in a context network, where inference rules are applied within and across those contexts.

The rest of this paper is organized as follows. Section 2 provides a high-level overview of our implemented mental state understanding system.

[2] Excerpt from "The Little Red-Cap" in Margaret Hunt's translation of the *Grimms Fairy Tales*. In some other translations, the story is also called "Little Red Riding Hood".

[3] Fauconnier (1985) covers a much broader set of space-builders including prepositional phrases ("from her point of view"), connectives ("if ... then ..."), and subject-verb combinations ("she thinks"). Our current system only deals with the last category.

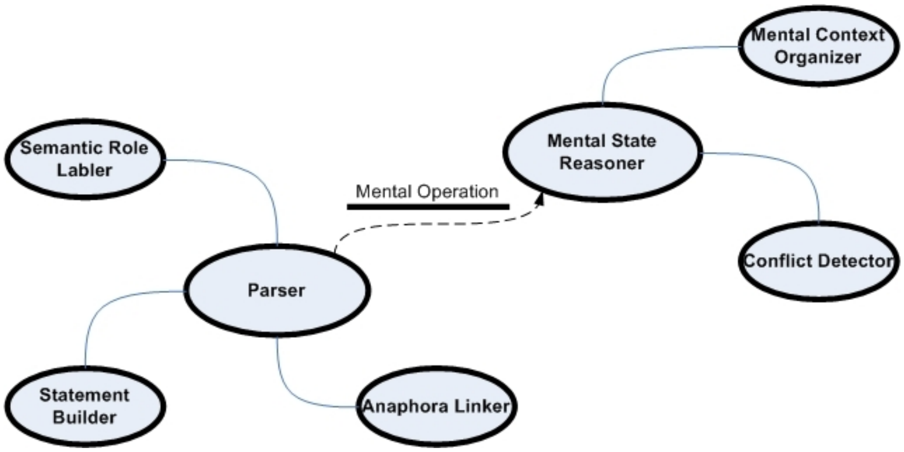


Figure 1: Overview of the mental state understanding system

Simple examples will be presented to demonstrate the input and output of the system. Section 3 and section 4 introduces our parser for mental state extraction and the inference engine for mental state representation, respectively. Section 5 discusses evaluation results on fictional and non-fictional children stories.

2 System Overview

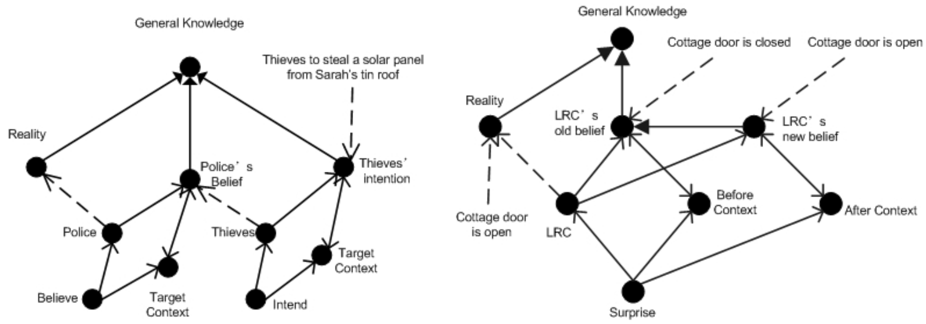
2.1 System Components

Similar to many story comprehension systems (e.g. [8]), our system consists of two parts. A parser combines several natural language processing components to extract useful information from raw text. The information is then integrated and filled into psych-term-argument templates to generate an intermediate semantic form called *mental operation*. The inference engine translates mental operations into a situation model represented by a semantic network. During this process, it fills in non-literal semantic information and maintains the situation model by a set of inference rules. Figure 1 shows the general structure of the system.

2.2 Example Output

To provide a general idea of what the system does, we temporarily treat it as a black box and use concrete examples to demonstrate its function. The input to the system is a piece of raw English text. The system processes one sentence at a time and outputs a situation model which represents the mental states of the characters appeared in the story.

Figure 2(a) and 2(b) show the semantic networks generated by our system given input sentences (1a) and (1b), respectively. As shown in the figure, our semantic representation of mental states is a set of mental contexts attached to different characters. Any inference or retrieval is done with respect to a specific context. Figure 2(a) shows a structured representation of the nested mental states in sentence (1a). We read the representation as: *In the police's belief, there is the thieves' intention, in which the thieves try to steal a solar panel from Sarah's tin roof.* Figure 2(b) shows a situation where little red-cap's mental image of the reality changes.



(a) The semantic representation of (1a). (b) The semantic representation of (1b).

Figure 2: The output models. Arrows with big solid heads represent "subcontext" relation. Arrows with dotted lines represent "in-context" relation. Arrows with solid lines represent "property" relation. Double-headed arrows represent "equals" relation.

According to these models, our system can generate and answer yes-no questions like[4]:

(2) a. Question: Do the police believe that the thieves were trying to steal a solar panel from Sarah's tin roof?
Answer: Yes.

b. Question: In reality, were the thieves trying to steal a solar panel from Sarah's tin roof?

[4] The questions are generated in the form of Scone language (a language used in the Scone knowledge-base), not English. We translate Scone language into English for the purpose of illustration.

Answer: Not sure[5].

c. Question: Before little red-cap was surprised, did she think that the cottage door was open?
Answer: No.

d. Question: Does little red-cap think the cottage door is open now?
Answer: Yes.

3 Integrated Parser

The parsing procedure consists of three stages: pre-processing, psych-term-argument parsing, and statement building.

3.1 Pre-processing

The pre-processing component consists of a sentence detector, a tokenizer, a chunker, and an anaphora annotator. We use the OpenNLP[6] toolkit to perform the first three tasks. Then we apply an in-house anaphora annotator to annotate pronouns in the text. The anaphora annotator implements a modified version of Strube's S-List and his coreference algorithm [9]. The S-List is a datastructure that maintains a list of sorted candidate entities for anaphora resolution. Following Strube, a set of ranking constraints is used for modeling readers' attentions. The constraints include the reader-old and reader-new discourse entitie labels ("brand new", "unused", and "evoked"). Our features used for anaphora resolution include number/gender agreements, binding constraints, types of nouns (proper or common) and partial string matching. Our implementation is mostly developed on children stories, in which there are many instances where the gender information for a proper noun is missing. For example, the program does not know little red-cap's gender the first time it sees the name in the story. But we allow that information to be filled in as the annotator reads the text. Also, some common features such as animacy and named entity class are not included in our feature set because they do not adapt well to this kind of narratives.

3.2 Psych-term-argument Parsing

The goal for the psych-term-argument parsing process is to produce flat mental operations (as opposed to structured mental operations which will be

[5] "Not sure" means "neither 'yes' nor 'no' ".

[6] http://opennlp.sourceforge.net/

generated by the statement builder). After pre-processing, the ASSERT [7] semantic role labeler is used to annotate each sentence with PropBank [6] argument labels. The most frequently used argument labels include TARGET (for psych-terms), ARG0 (for agents), ARG1 (for patients or propositions), ARG2 (for patients or propositions when ARG1 is absent), ARG-MOD (for modal verbs) and ARG-NEG (for negations).

Although every sentence is processed, the system only looks for those containing the pre-defined psych-terms. The set of psych-terms were extracted automatically from the WordNet along the synset (sets of synonyms) links. The seed words used for collecting mental expressions from the WordNet contains 10 mental verbs: “think”, “want”, “pretend”, “confess”, “surprise”, “satisfy”, “frustrate”, “decide”, “regret” and “worry”. For each mental expression returned by WordNet, we restrict the next search depth to 3. Using this method, WordNet returns 239 verbs and phrases, among which we choose 42 psych-terms that are relatively less ambiguous for our initial system development. The 42 psych-terms also include modal verbs such as “will” and “must”. These psych-terms are matched to the text annotated by ASSERT. Parses like (3a) and (3b) are then extracted[7].

(3) a. 0: [ARG0 The police] [TARGET believe] [ARG1 the thieves were trying to steal a solar panel from Sarah 's tin roof]

b. 0: The police believe [ARG0 the thieves] were [TARGET trying] [ARG1 to steal a solar panel from Sarah 's tin roof]

A set of grammar rules are used to map the target verbs and their arguments to mental operations like (4a) and (4b). At the implementation level, the mental operations are Lisp functions to be evaluated by the mental state inference engine.

(4) a. (new-single-modal {The police} '((new-statement {the thieves were trying to steal a solar panel from Sarah 's tin roof})) {belief})

b. (new-single-modal {the thieves} '((new-statement {to steal a solar panel from Sarah 's tin roof})) {intention})

3.3 Building Statement Structures

The psych-term-argument parsing components generate a set of mental operations. However, not all of these operations ought to be evaluated by the

[7] We have 2 statement checking rules to correct possible ASSERT output errors on propositions.

inference engine. In general, mental operations generated from complementizer phrases should not be evaluated. Instead, they should be passed to their parent operations as proposition arguments. For example, mental operation (4b) comes from the complemetizer phrase "the thieves were trying to steal a solar panel from Sarah's tin roof". If this operation is evaluated in the inference engine, we will not get the same semantic representation as in Figure 2. Instead, the system will erroneously judge the proposition "the thieves were trying to steal a solar panel from Sarah's tin roof" to be true in reality, while the correct representation is to make it true in the police's belief. To avoid such errors, (4b) is passed to its parent mental operation (4a) and made an argument of (4a), which ensures that (4b) will be evaluated under the police's belief. The product of this process is a structured mental operation (5). And finally, only (5) is sent to the inference engine. In summary, the goal of the statement builder is to build and evaluate the correct mental operations.

(5)

(new-single-modal {The police} '((new-single-modal {the thieves} '((new-statement {to steal a solar panel from Sarah 's tin roof})) {intention})) {belief})

To build a structured mental operation, each candidate mental operation is stored in a list. A topological sort is performed based on the complementizer phrase relationships among different mental operations. Such relations are found through argument span check. That is, if some arguments of operation 1 are all found in operation 2's proposition argument, we will assume operation 1 is operation 2's child operation. After the topological sort, the head of the list stores the element mental operations which do not have children operations, and the tail of the list stores the mental operation that covers the whole sentence. Then the operations are fed into their parent operations one by one along the list. Finally, those operations that have no parents are sent to the inference engine.

4 Mental State Inference Engine

In this section, we briefly explain the structure and mechanisms of the mental state inference engine.

4.1 Context Activation Mechanism

The mental state inference engine is built on top of the Scone knowledge-base (KB) system. Scone is designed to be a practical KB with emphasis on its efficiency of the most commonly used operations for search and inference. Regarding these goals, Scone provides default reasoning with exceptions. As we have shown, Scone can be viewed as a semantic network representation, with nodes representing entities and links representing relations or statements tied to these entities. At a higher level, the types in Scone may be viewed as frames or descriptions. A multi-context and context activation mechanism is designed into Scone using the marker-passing algorithm [3]. In this paper, a context is used to represent the state of mental attitudes. In Scone, the context nodes are treated as the other nodes in that they are also tied into their own inheritance hierarchy using "sub-context" links. However, the "sub-context" relation represented by inheritance is a mechanical one. The relation between the two contexts is neither "is-a" nor "part-of", but something more like "is a clone of, modulo explicit changes". Contexts can also be used to represent the state of a changing world at different times. Each of the contexts represents a mental attitude at a specific time; it begins as a clone of the state in the previous time-step, and then we can add or remove a few items if necessary.

4.2 Mental Context Representation

In general, the mental context model tracks the changes of the mental state [2]. At each time point, it builds a mental context network that represents nested mental states. The input to the model is a list of mental context operations extracted from text. Each of the operations corresponds to one psych-term. The complex semantics of a psych-term is factored into a set of atomic operations on single mental contexts through semantic decomposition. These contexts are organized into a hierarchical structure, which constitutes a simplified representation of human memory.

The semantics of psych-terms are projected onto the context network through semantic decomposition[8]. For example, one sense of the word "pretend" can be represented as "X is not present in reality and person P1's belief, but P1 wants person P2 to believe it" (these semantic definitions are restricted to mental contexts). This example involves several contexts: the reality, the belief of P1, the intention of P1, the belief of P2 under the intention of P1, as well as the before context and the after context of "pre-

[8] Recent work on verb entailment can be found in [5].

tend". Note that the mental operations are higher order, so there can be other psych-terms (e.g. "want") embedded the definition of "pretend".

Mental operations update the mental context network in two aspects. First, they build a context inheritance chain which represents the evolution of a single mental context at different time steps. For example, in the "Little Red-Cap" story, little red-cap has different belief contexts at different time. By default, each of the newly updated versions of one's belief would inherit from his/her most recent belief. Second, the mental operations are used to build a hierarchical context structure which organizes multiple types of mental context according to events and agents. Figure 2 illustrates three basic aspects of the context structure:

1. By default, the mental contexts inherit from a general knowledge context which represents the general background knowledge of a story.

2. Mental contexts can be organized by events. A typical mental event has an agent whose mental contexts are changed as an effect of the event. Each psych-term would be mapped to one of the mental events. When we retrieve an event, a set of related contexts of that event would also be retrieved.

3. The mental contexts are environment-sensitive. For example, the thieves' real intention can be different from the police's belief of their intention.

In our representation, different instances of the mental contexts are organized in a dynamic context structure. We could then constrain the behaviors of different mental contexts under different mental events using inter-contextual rules. Once a mental event (e.g. "little red-cap was surprised") triggers, the related mental contexts would check and modify their own structures and contents based on the new information. Usually this self-adjustment can be achieved by a realization of a difference between the external world and the belief, assumption or expectation. According to this, newly updated mental contexts would be constructed.

5 Evaluation Results

We use 513 children stories from Project LISTEN's[9] reading tutor story database for system evaluation. This corpus contains 213 fictional articles

[9] http://www.cs.cmu.edu/l̃isten

and 300 non-fictional (or informational) articles. From these, our system extracts 1181 mental state expressions in fictions and 413 in non-fictions. After the parsing stage, 60.80% of the fictional and 62.71% of the non-fictional mental state expressions are fully parsed (i.e. there is no empty arguments for the mental operations) and sent to the inference engine. After processing of the mental operations, the system automatically generates 1537 yes-no questions and answers for fictional texts, and 518 for non-fictional texts. The question-answer pairs are generated by traversing newly visited mental contexts and statements/objects bundled in those contexts immediately after each mental operation is evaluated. These questions and answers are all in similar forms as the examples demonstrated in section 2.2.

We do a careful evaluation on 431 questions for fictional texts and 155 for non-fictional text that are randomly selected from the question-answer pool. Since both the questions and answers come from the situation model stored in the system, an error occurred in either the question or the answer counts for an incorrect example.

Table 1 shows the error rates of each error category for both fictional and non-fictional stories. The second column ("argument error") gives the percentage of incorrect question-answer pairs caused by wrong arguments in the mental operations. This type of error comes from a misinterpretation of the ASSERT output. For example, we assume that ARG1 indicates a proposition for psych-terms if it is not a noun phrase. But "she [TARGET wanted] [ARG1 so much]" is an exception of this assumption. The third column ("statement error") gives the percentage of incorrect examples (usually questions) caused by incomplete or unnecessary statements in the mental contexts. This type of errors results from inaccurate ASSERT outputs that are not captured by our statement checking rules. Space-builder error (the fourth column) refers to the cases in which the mental contexts are not correctly constructed (this results in incorrect answers). This usually happens when there is a mental space that has been neglected by our system. For example, when given sentence "Dig for it if you want the gold", our system would only look at "you want the gold" and treat it as a valid statement, without noticing that it is embedded in an if-clause which indicates another mental space. Ambiguity error (the fifth column) refers to errors cause by the lexical ambiguity of psych-terms (e.g. the word "will" indicates future tense in "Pilly will go to school tomorrow", but not in "Some sharks will eat just about anything"). Negation error (the sixth column) occurs when there is a negation that has been neglected (this results in an incorrect answer). For example "It will do him no good, neither will it help anybody else" means "It will not help anybody else". But this negation is

Table 1: Evaluation Results

	Argument Errors	Statement Errors	Space-Builder Errors	Ambiguity Errors	Negation Errors	Total Errors
Fictions	6.26%	8.12%	8.35%	4.64%	4.64%	26.68%
Non-Fics	1.29%	2.58%	7.10%	7.74%	1.29%	19.35%

not captured by our system. Note that since the five types of errors listed in Table 1 are not mutually exclusive, the error rates in each row of the table do not sum up to the total error rate of question-answer pairs generated by the system. Anaphora error has been counted separately since this type of error alone has a significant effect on system performance, and the anaphora annotator is relatively independent with our task compared to the other in-house components. During evaluation, we only apply anaphora resolution to pronouns (the word "it" is not counted). In the same question-answer pool, we observe the anaphora error rate of 19.49% on fictions and 17.42% on non-fictions.

6 Conclusion

This paper presents an implemented system that understands mental states expressed in narratives. The system extracts and processes mental states by mapping psych-terms and their arguments to mental contexts stored in a situation model. The system is evaluated by automatically generated question-answer pairs. Future directions include extending the system to a broader set of psych-terms and patterns to cover more mental states from natural language. Meanwhile, we will explore the two-way interaction between the parser and the inference engine to refine the overall processing accuracy.

7 Acknowledgements

The author would like to thank Scott Fahlman and Jack Mostow for many helpful discussions. The research reported here was supported in part by the Institute of Education Sciences, U.S. Department of Education, through Grant R305B070458 to Carnegie Mellon University. The opinions expressed

are those of the authors and do not necessarily represent the views of the Institute.

References

[1] Afzal Ballim and Yorick Wilks. Beliefs, stereotypes and dynamic agent modeling. *User Modeling and User-Adapted Interaction*, pages 33–65, 1991.

[2] Wei Chen and Scott E. Fahlman. Modelling mental context and their interactions. In *AAAI Fall Symposium on Biologically Inspired Cognitive Architectures*, 2008.

[3] Scott E. Fahlman. Marker-passing inference in the scone knowledge-based system. In *KSEM'06*. Springer-Verlag, 2006.

[4] Gilles Fauconnier. *Mental Spaces: Aspects of Meaning Construction.* MIT Press, Cambridge, MA, USA, 1985.

[5] Lauri Karttunen. Word play. *Computational Linguistics*, 33(4):443–467, 2007.

[6] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005.

[7] Sameer Pradhan, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky. Support vector learning for semantic argument classification. *Machine Learning*, 60(1-3):11–39, 2005.

[8] C. K. Riesbeck and R. C. Schank. Comprehension by computer: Expectation-based analysis of sentences in context. Technical Report 78, Yale Computer Science Department, 1976.

[9] Michael Strube. Never look back: an alternative to centering. In *Proceedings of the 17th international conference on Computational linguistics*, pages 1251–1257, Morristown, NJ, USA, 1998. Association for Computational Linguistics.

[10] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. In *Language Resources and Evaluation*, volume 39, pages 165–210, 2005.