# Unsupervised learning of common sense event structures from simple English stories

Ken Arnold and Dustin Smith

June 6, 2008

## Abstract

We present a program that builds up an internal event representation by exposure to common sense stories. The stories come from the OMICS corpus, and are English narratives about the steps involved with everyday household tasks, such as "get the mail" and "make a bed". With repeated exposure to different stories describing the same events, our program learns a sequential hierarchical structure that reveals the sequential nature of events and the nested relationship between events (goals[1]) and sub-events (sub-goals). Single events are merged using *lexical semantic equivalence*, and both single events and event chains are merged using *functional equivalence*. Functional equivalence of an action means both its preceding and succeeding events are equivalent.

Section 1 of this paper provides an introduction to the problem of giving computers common sense, which leads into a multidisciplinary exposition of the event chain or "narrative" representation. Section 2 explains the working of the program, and describes how it relates to the problem of concept learning. Section 3 concludes with a conclusion of contributions, the problems of this approach, and next steps for this research.

## 1 Introduction

How do we get knowledge into computers? Artificial Intelligence researchers quickly came up with the idea that knowledge could be acquired the same way as people do—by learning from text: reading.

Machine reading, however, is fraught with many technical challenges. To computers, natural languages merely form strings of words. To people, these words are just the "tip of the iceberg" of the text's underlying meaning. But extracting the underlying meaning generally requires volumes

---

[1]The term 'goal' means something more specific than 'event'. It implies that this knowledge is part of a plan from the perspective of an agent that is executing the plan. In other words, goals are a subset of events; and the analog from the field of cognitive science is: procedural memory is a subset of episodic memory.

of background knowledge, what we call **common sense**, most of which is shared between almost all humans but so far inaccessible to computers.[2]

The result is a bootstrapping problem. When people read, they learn by connecting new knowledge to what they already know [Kay, 1982], and a lot of knowledge is already implicit behind the text. Starting with some base of knowledge, a machine reading program could infer other knowledge, but understanding even childrens' stories often requires large amounts of background data. So, how do we get the *common sense* knowledge into computers?

## 1.1 Acquiring commonsense knowledge in natural language

It is rude and costly to tell people what they already know, so most of the basic knowledge of the world is *too obvious* to ever be written. Yet there is a glut of common sense knowledge harbored and concealed in the minds of many people; it, by definition, is obvious and known to nearly everyone.

The Internet provides an unprecedented potential for accessing human intellectual resources. Realizing this, Push Singh and Hugo Liu and started the Open Mind Common Sense (OMCS) knowledge acquisition project [Singh, 2003]. OMCS is a website where volunteers can contribute simple world knowledge in the form of assertional sentences in their native language[3]. A philosophy behind the OMCS is to use natural language as a knowledge representation, instead of one of the many man-made representations that flourish in the artificial intelligence and linguistic communities. (Compare this to the alternative approach of the CYC project that approaches the same common sense problem by encoding knowledge in a formal predicate calculus-like representation to accommodate automatic inference [Matuszek et al., 2006].)

The long term goal of the OMCS project is to develop a resource for machine reading and learning from text—a system that reads and bootstraps its knowledge to learn from increasingly more 'semantically compressed' text.

## 1.2 Representational issues: From English to ...?

Processing and *using* the English common sense assertions is another problem. The current corpus of ~800,000 assertions can readily be transformed into a semantic network with 18 binary predicates of the form: `IsA(dog,animal)`. This semantic network incarnation is called CONCEPTNET. While very useful, this representation has major problems:

- **Lexical semantic underspecification.** Even the common sense statements themselves require *other* background knowledge to be understood, because they often assume a context. For example [Minsky, 2006], the knowledge "a person does not want to get wet" does not explain the contexts of when people are drinking fluids, taking a shower, or swimming. There

---

[2]In developing ADVICE TAKER[McCarthy, 1959], an attempt to give computers the ability to learn from reading, John McCarthy was the first to recognize that such a program would require background knowledge, i.e., common sense.

[3]The current version of OMCS supports several languages in addition to English: http://commons.media.mit.edu

are underlying meanings behind the situation that specify *where* a person does not want to get wet (on the clothes that they are currently wearing) and *why* they do not want to get wet (wearing soggy clothes is uncomfortable), which in aggregate may form the majority of the *getting wet* episodes to merit this statement.

- **Not enough structure to perform reliable inference.** The assertional structures are natural language, and the structure (needed for inference and learning) is limited to 18 binary relationships. No type restrictions are placed on either concept, and ambiguity and noise means that you don't know what "kind" of concept will be on the other side of the semantic link you are considering. The X in `IsA(dog,X)` could be either a category, `'pet'`, or some more expressive statement, like: `'threat to the sanitation of urban parks and sidewalks'`.

- **Retrieving knowledge relevant to the problem.** Different problems require different pieces of knowledge. Due to the aforementioned structural limitation, CONCEPTNET suffers the problem of not being able to retrieve *only* the knowledge relevant to the given problem. If we want to know *where* to find a dog, we can formulate this as `LocationOf(dog,X)`; however, this does not help us, for example, when solving the problems of (a) locating a place to buy a hot dog, or (b) knowing where you can buy a new pet. The current mechanisms for retrieving knowledge (text searching by keyword) have reasonable recall but terrible precision: the relevant answers are diluted by a large number of unrelated assertions.

- **Independent assertions.** The knowledge is only connected by nodes that share the same lexical expression. The concept "eat" connects all knowledge about eating. It does not distinguish dogs eating from humans eating; eating at a restaurant from eating at home; lunch from dinner; and so forth. Taken all together, the knowledge is full of contradictions and inconsistencies, because there is missing contextual information from the assertions that modulate the relevance of the eating concepts to the given situation.

The above list of grievances indicates a common problem: Assertional knowledge alone is insufficient; the system requires meta-knowledge about what knowledge is relevant to a given situation and goals if it is to use the knowledge in predicting, planning, learning, evaluating or some other such cognitive function. Obviously, knowledge is useless unless it can be used to solve problems; so, what better way to represent the knowledge than within a problem-solving context—what is known to the AI communities as a *plan* or *script*—that we will call **event structures**.

## 1.3   The event representation

The event representation has many merits, including:

- **Sequential structure**. The sequential structure of the narrative describes changes as they occur over space and time, reflecting the sequential nature of reading.

- **Constraining inference**. Inference is a combination of the reader's background knowledge and the knowledge that is explicitly in the text. To keep both reader and author "on the same page", inferences should be relevant to the topic at hand.

- **Inferential tractability in a large knowledge base.** Unconstrained natural language has a very broad scope. For understanding to occur, however, stories must follow along a path shared between the reader and author.

- **Representing context**. The missing information of the statements could come from different parts of the narrative representation. For example, in the script of "getting mail", when you proceed from "find mailbox" to "open mailbox", it is already obvious *which* mailbox you are opening (the one you found earlier) and *why* you are opening it (to get mail).

And has been gaining traction from many research fields:

- **Artificial Intelligence**

  - **Cognitive architectures** Marvin Minsky has proposed the *Emotion Machine* cognitive architecture [Minsky, 2006] with an intricate system of knowledge organization, that emphasizes the role of reflection (metacognition) and organizing knowledge in sequential story-like scripts. These scripts are indexed by contextual meta-knowledge that include the agent's goals. Push Singh implemented this planning/meta-planning system [Singh, 2005] using script-like cases called "commonsense narratives."

  - **Knowledge engineers**. Large-scale common sense knowledge bases, like CYC and OMCS, have realized that logical consistency is too brittle for common sense reasoning. Strategies such as non-monotonic representations have been developed to bypass the obvious problems (`All birds can fly.  Penguins are birds and can not fly.`). Instead, this knowledge could be represented by the event structure, where exceptions live in different, isolated events. CYC's logical representation is only consistent within a small module, called a *microworld*, because its developers found global consistency too cumbersome.

  - **Inference**. Well-studied inference rules like *deduction* can guarantee sound conclusions from sound premises; however, analogical reasoning is more powerful (deduction can be thought of as a subset of analogical pattern-matching rules) but requires a richer structural components. Event representations can provide a structure causal information, where implicit knowledge propagates along the sequence.

- **Psychological Sciences**

  - **Memory studies**. The early studies of psychologist Charles Bartlett [Bartlett, 1995] demonstrated that people are able to better remember stories if they are consistent

with a familiar narrative arch. In Roger Schank's recent theory of human memory [Schank, 1995], he has proposed that all knowledge consists of stories, and cites many of its merits and some research anecdotes in support of this hypothesis.

– **Causal Schemas**. A popular representation in cognitive science is the relational schema that represents relations between items in a particular domain of knowledge. Josh Tenenbaum and colleagues have demonstrated techniques for learning schemas from data [Kemp et al., 2006], but it is still not clear how many schemas there are, and how people retrieve them to solve problems, and when one schema ends and the other begins. Clearly, the causal relationships of a event structure can be related to the notion of schema.

- **Linguistics**. Many theories in lexical-semantics have posited that sentences convey meaning by referencing semantic event structures. Predicates (usually verbs) accept arguments that often describe some sort of change to these arguments. Beth Levin suggested that these underlying events affect the syntactic organization of sentences, and exhaustively listed the combinatoric possibilities of ordering verb predicates and their arguments, donned the awkward sentences with an '*', and then clustered the underlying verbs by their asterisks [Levin, 1993]. As there is no one-to-one mapping between sentences and the underlying event structure, there are many different theories about kinds of semantic event frames and how they influence the roles of the sentences' constituents [Gildea & Jurafsky, 2002].

- **Philosophy**.

   – **Concept Acquisition**. In Fodor's survey, he demonstrated that similar concepts seem to be defined by their local properties, because global definitions some times are impossible; for example, when trying to define the property of "bird", you would be hard-pressed to find a description that included: an ostrich, a dead bird, a drawing of a bird, etc. Fodor also dismisses the pragmatic interpretation of concepts: concepts are defined by their ability to be *used* in sorting tasks. His objection is that some concepts are singletons (*e.g.*, the concept of evolution) and have no related things to be sorted against. A reasonable alternative hypothesis would be to making the pragmatic definition more general: namely, allowing concepts to be the mentalese (however represented) that is used to perform a larger range of cognitive functions (sorting, prediction, identification, inference). In our project, we make use of this "functional" definition of concepts, where one event "concept" is the same as another if it is exchangeable within the event sequence.

   – **Induction**. The learning process of inferring that a particular property or relation is relevant to a large class after observing only a particular instances is known as *induction*. Philosophers like Nelson Goodman have realized that induction alone needs to be constrained to learn a lot from so few examples. An event structure that represented

actions as changes of properties and relations could be used to determine how to match the properties with their causal influences to constrain/generate the inductive hypothesis space.

### 1.3.1   Generating event structures from common sense scripts

With the goal of learning event chains, we need some simple corpus of common sense scripts. Many researchers have made small steps toward interpreting rich non-common sense text.

How did we get this corpus of common sense scripts? Honda corporation set up a clone of OpenMind project, called the Open Mind Indoor Common Sense OMICS project, to collect *indoor* knowledge, namely, knowledge about common household items and activities. OMICS also represents knowledge in English, in addition to assertions, it includes a corpus of common house-hold tasks. For example, here are the first four of 52 stories about the task: `sweep floor`:

```
1.a locate broom
1.b pick up broom
1.c run broom across floor, gathering dust into a pile
1.d throw away the dust and dirt

2.a locate broom
2.b run broom across floor to gather dust into a pile
2.c place dust pile in trash can

3.a take broom
3.b move broom back on forth
3.c move garbage towards some location
3.e pick up garbage and throw away

4.a get the water
4.b splash the water on the floor
```

If we make the assumption that all of this knowledge is correct and refers to the same underlying event (cleaning the floor), we can treat these 52 stories as many opportunities to begin to learn a richer event structure. The example illustrates several challenges that this task will face:

1. **Multiple ways to say the same thing.** 1.d and 2.c talk about the same activity (throwing away debris) in different ways. Language is flexible, so this is no surprise. To match up these alternate descriptions, our system uses WordNet [Miller et al., 1990] similarity metrics.

2. **Descriptions differ as to their levels of detail.** Another problem (central to the common sense problem) is that a description of one task often contains several sub-tasks. In AI, a common representation for these types of plan is a hierarchy (*e.g.*, [Hoang et al., 2005]). As an example (more compelling examples can be found in other stories), step c of stories 1 and 2 could be considered a concise description of the two-steps 3.b and 3.c. To accomplish this, we grouped repeating sub-sequences of events that were **functionally equivalent** (defined in section 2.4).

3. **Overlapping context**. Each event assumes some underlying context and it is hard for people to articulate where on the contextual umbilical cord the mother (preceding event) ends and the baby (target event) begins. Quite often there will be contextual overlap. For example, the author of story 3 assumed that the location of the broom was already known, while the authors of stories 1 and 2 included the sub-task of "locate broom".

4. **Descriptions have causal discontinuation.** While stories 1—3 describe the process of "sweeping the floor", story 4 is describing the related chore of "mopping a floor". A more advanced learner could recognize that these are different enough that they may be different events.

5. **Parsing problems**. The statements are basic English imperatives, but most NLP tools are designed for processing long, complete, *Wall Street Journal*-style sentences. Hence, it will be difficult to extract predicate-argument structures automatically.

## 2    Implementation

Our narrative learner reads user-contributed stories from the OMICS actions database. For each task (e.g., "make tea", "answer the phone", "get the mail"), it constructs a hierarchical script representation that summarizes the different ways to describe or perform various subactions.

### 2.1    Parsing

The stories are given in natural language (English), in a terse imperative form. However, for automated processing, we needed the stories to be in a structured representation. So the stories are first parsed into a simple verb-argument form. The following table shows example parsing results for a "get mail" story:

| Input Story | Parsed Story |
|-------------|--------------|
| go to mailbox | go\|mailbox |
| open mailbox | open\| mailbox |
| retrieve mail | retrieve\|mail |
| close mailbox | close\|mailbox |

Table 1: Parsing results for the first get_mail story

This parsed representation is clearly limited, but its focus on the verbs used helps the system to learn how actions are grouped.

## 2.2   Graph Representation

Each task is represented as a directed acyclic graph connecting the start of the task with its conclusion, representing the possible steps to take to accomplish the action. Each node in the graph represents a sub-task, which itself may be a graph. The graph is built up one story at a time. Here are the graphs for the first two "get mail" stories:
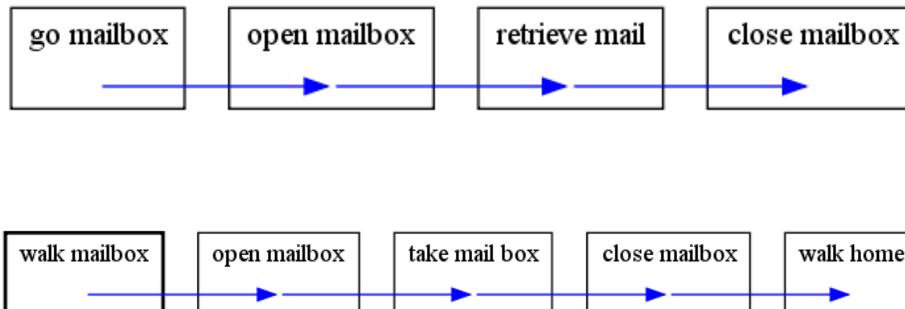




Each new story is first added to the graph as an entirely separate path between the start and end nodes for the task. Not even exactly identical actions are necessarily unified, since it is possible that the implicit context differs. The system then makes two passes to combine the nodes of the graph when it has evidence that certain nodes should be grouped. The first pass simply joins any nodes that are considered equal. (The subtleties of equality will be discussed in the Action Equality section.) The following figure shows the result of this first merge operation on the two stories shown previously:
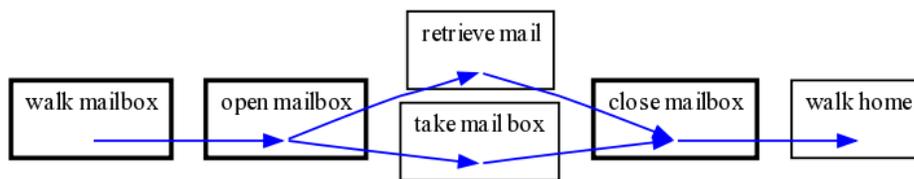


Plan structure for 'start story' (n=1, pre)

The graph shows that the two stories are nearly identical, except that the middle action can be described by "take mail out of box" or "retrieve mail". The graph structure gives evidence that these nodes are *functionally equivalent*: they are what happens between the same two other steps: opening the mailbox and closing it. The second merge pass finds these sorts of partitions in the graph and groups them into a node representing the combined action. Specifically, we look for two nodes A and B in the graph ("open mailbox" and "close mailbox" in the example) connected by a set of paths P such that:

9

1. The graph contains two or more paths from A to B (in the example, through "take mail box" or "retrieve mail")

2. At least one of the paths passes through just one intermediary node. That action is a potential name for the combined action, since the other actions are potentially breaking that action into several steps. In the example, both paths qualify.

3. None of the paths may be a direct connection. In that case, the step might have been optional.

When several possible node pairs are found, we process the one with the shortest total path length first.

If such a node pair is found, the system creates a new action node representing all of P and replaces P by the new node. Ideally no edges connect an intermediary node in P with some node outside of P, but if this does happen, the combined node is substituted for the intermediary node. These edges could also be used to evaluate the quality of the grouping, but for now they simply result in a warning.

The entire merge process is repeated until no additional nodes are merged.

The following figure shows the graph after the second merge pass:

Plan structure for 'start story' (n=1, post)
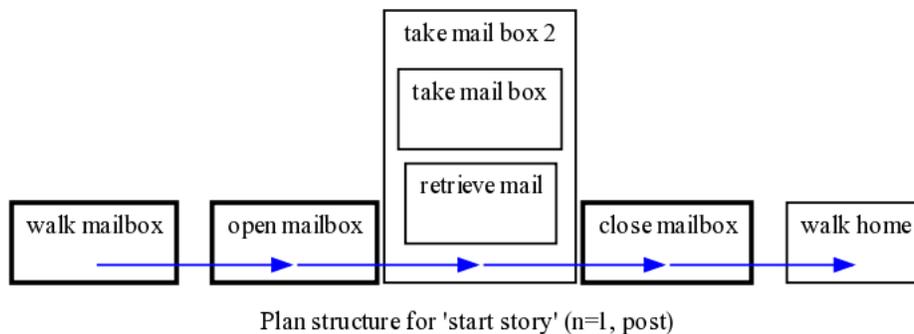
The paths are computed using the Floyd-Warshall algorithm modified to record all paths (instead of the shortest path).

## 2.3  Action Equality

The first merge pass merges semantically equal nodes. But how do we determine that "go to the mailbox" and "walk to the mailbox" are (basically) semantically equivalent? Besides exact verb/argument equality, we also treat two nodes as identical if their Wu-Palmer similarity exceeds 0.85[Mihalcea et al., 2006]. This metric worked well in many cases, but leads to problems when WordNet does not adequately capture various semantic distinctions. For example, "go inside" and

"go outside" have a Wu-Palmer similarity of 0.96, despite their drastic difference in meanings[4], causing the algorithm to fail to parse many stories (the false similarity creates cycles in the graph). Future improvements could include evaluating similarity with respect to the context (e.g., whether the destination or the mode of travel is important).

## 2.4 Evaluation

When it completes successfully, the system learns event structures that look reasonable, capturing functional equivalence and hierarchical subtasks. We did not attempt formal validation because of the challenge of determining a suitable reference result and a suitable metric for comparing event structures.

However, the motivation of the project is not simply to determine event structures, but to understand other stories. Thus a metric of, for example, how well the system predicts the next event in a new story or identifies an anomaly might be more meaningful.

## 2.5 Building the corpus

1. Obtained the Indoor Commonsense Database from Honda[5].

2. Generated a corpus from the 50 tasks with the most independent scripts with lowest variance.

3. Parsed each step into a verb+arguments structure using the TAGCHUNK joint part of speech tagger and syntactic chunker [Daume III & Marcu, 2005]. We simply treat the first word as the verb (since all steps are imperative) and all the noun phrases as arguments.

# 3 Future Directions

## 3.1 Active Learning and Inference

The whole point of learning an event structure is to be able to guide inference during machine learning. **Inference** simply means, quite generally, the process of deriving new knowledge from existing knowledge. Listen to Peter Norvig describe why people are motivated to read:

> "There is an implicit contract between the author and reader wherein the author agrees to explicate, enough of the situation so that the reader, by searching for proper inferences, can recover the rest of the information and make sense of the text. ... When these connections are missing, or when inferences prove to be incorrect, it is usually a signal that the writer is being humorous, ironic, mysterious, has a different view of the

---

[4]Though in some ways the two actions are very similar: they both involve a change of location toward a general type of area of the world. Similarity, then, cannot be one-dimensional.

[5]Freely downloadable as a database here: http://openmind.hri-us.com/

world, or is just being confusing. Indeed, much of what makes texts interesting is the intentional flaunting of this implicit contract. [Norvig, 1989]

Our future system should more actively engage in the learning process in an iterative fashion. Currently, its learning mechanics are part of its parsing. Instead, it should be part of an architecture that has a ⟨Predict⟩ → ⟨Observe⟩ → ⟨Compare⟩ → ⟨Reformulate⟩ iteration.

## 3.2   Better Algorithms

The graph-manipulation algorithms we employed, while motivated by the theory of event descriptions, are overly simplistic. For example, better hierarchy-learning algorithms would enable the system to re-evaluate decisions it made about how events group together if new evidence suggests that the original opportunistic grouping was incorrect.

## 3.3   Incorporating Context

Even in these simple stories, significant and useful context is often missing (e.g., "I know where the broom is." or "There is mail in the mailbox."). Some of this context could be automatically extracted based on lexical semantic (common sense) knowledge about the verbs involved.

## 3.4   More Realistic Input

The narratives we used for this stage of the project are very simple in structure: they have a single protagonist, simple imperative language, and little in the way of deliberation. The system would fail dramatically on all but the simplest children's book, for example. The system should be able to handle descriptions of situations at a level of detail closer to our day-to-day experience.

# 4   Related Work

People have worked on this problem before, but often from more complicated, non-common sense text [Chambers & Jurafsky, 2008, Mooney & DeJong, 1987, Mooney, 1990, Berwick, 1983, Livingston & Riesbeck, There are two main problems with this:

1. The systems do not have enough repeated examples (it is rare to have a resource of multiple of the same story told different ways).

2. The material is too condensed. The story that is being parsed is a single path through an *enormous* planning space. Hopefully, the common sense scripts from OMICS are at a much finer description level and have fewer degrees of freedom/are more complete.

# References

[Bartlett, 1995] Bartlett, F. C. (1995). Remembering: A study in experimental and social psychology. (pp. 337). 4

[Berwick, 1983] Berwick, R. C. (1983). Learning word meanings from examples. *International Joint Conference on Artificial Intelligence*, 1. 12

[Chambers & Jurafsky, 2008] Chambers, N. & Jurafsky, D. (2008). Unsupervised learning of narrative event chains. *ACL-08*, (pp.9̃). 12

[Daume III & Marcu, 2005] Daume III, H. & Marcu, D. (2005). Learning as search optimization: approximate large margin methods for structured prediction. *ACM International Conference Proceeding Series*, 119, 169–176. 11

[Gildea & Jurafsky, 2002] Gildea, D. & Jurafsky, D. (2002). Automatic labeling of semantic roles. *Association for Computational Linguistics*. 5

[Hoang et al., 2005] Hoang, H., Lee-Urban, S., & Muñoz-Avila, H. (2005). Hierarchical plan representations for encoding strategic game ai. *Proc. Artificial Intelligence and Interactive Digital . . .*. 7

[Kay, 1982] Kay, P. (1982). Three properties of the ideal reader. *eric.ed.gov*. 2

[Kemp et al., 2006] Kemp, C., Tenenbaum, J. B., Griffiths, T., Yamada, T., & Ueda, N. (2006). Learning systems of concepts with an infinite relational model. *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, (pp. 381–388). 5

[Levin, 1993] Levin, B. (1993). English verb classes and alternations: a preliminary investigation. *books.google.com*. 5

[Livingston & Riesbeck, 2007] Livingston, K. & Riesbeck, C. (2007). Using episodic memory in a memory based parser to assist machine reading. *Working notes, AAAI Spring Symposium on Machine Reading*. 12

[Matuszek et al., 2006] Matuszek, C., Cabral, J., Witbrock, M., & DeOliveira, J. (2006). An introduction to the syntax and content of cyc. *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, (pp. 44–49). 2

[McCarthy, 1959] McCarthy, J. (1959). Programs with common sense. 2

[Mihalcea et al., 2006] Mihalcea, R., Corley, C., & Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. *Proceedings of AAAI*, 6. 10

[Miller et al., 1990] Miller, G., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1990). Introduction to wordnet: An on-line lexical database*. *International Journal of Lexicography*, 3(4), 235–244. 7

[Minsky, 2006] Minsky, M. (2006). *The Emotion Machine.* 2, 4

[Mooney, 1990] Mooney, R. (1990). Learning plan schemata from observation: Explanation-based learning for plan recognition. *Cognitive Science.* 12

[Mooney & DeJong, 1987] Mooney, R. & DeJong, G. (1987). Learning schemata for natural language processing. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, (pp. 681–687). 12

[Norvig, 1989] Norvig, P. (1989). Marker passing as a weak method for text inferencing. *Cognitive Science.* 12

[Schank, 1995] Schank, R. C. (1995). Tell me a story: Narrative and intelligence. (pp. 253). 5

[Singh, 2003] Singh, P. (2003). The open mind common sense project. 2

[Singh, 2005] Singh, P. (2005). Em-one: An architecture for reflective commonsense thinking. 4