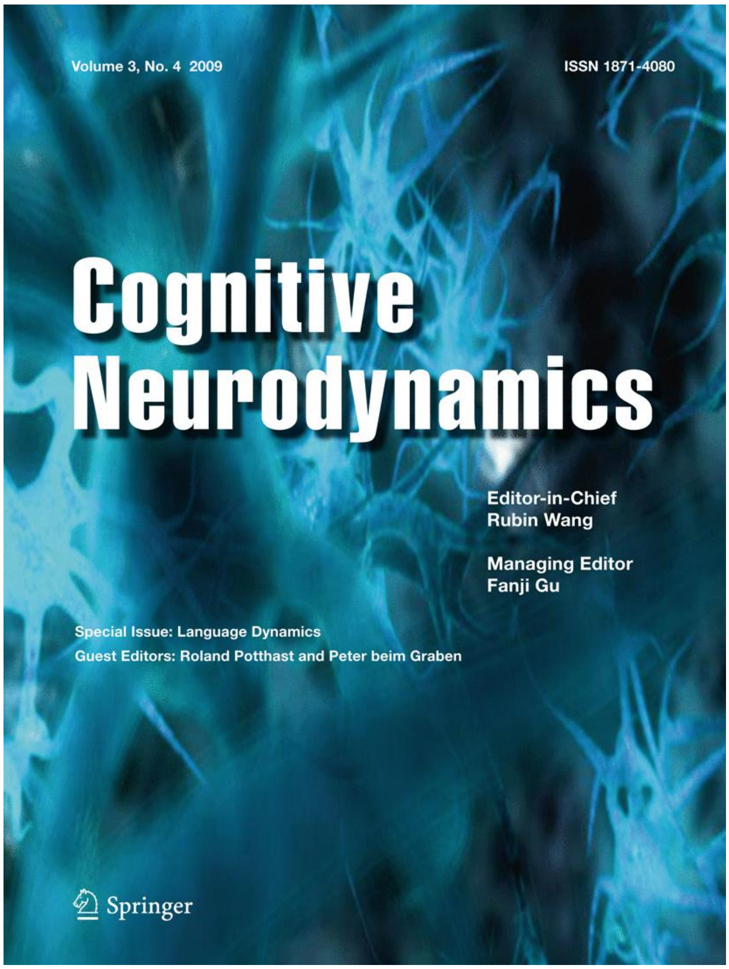
ISSN 1871-4080, Volume 3, Number 4



Cogn Neurodyn (2009) 3:365–372
DOI 10.1007/s11571-009-9097-x

RESEARCH ARTICLE

Memory reconsolidation for natural language processing

Kun Tu · David G. Cooper · Hava T. Siegelmann

Received: 2 April 2009 / Revised: 30 September 2009 / Accepted: 10 October 2009 / Published online: 28 October 2009

Abstract We propose a model of memory reconsolidation that can output new sentences with additional meaning after refining information from input sentences and integrating them with related prior experience. Our model uses available technology to first disambiguate the meanings of words and extracts information from the sentences into a structure that is an extension to semantic networks. Within our long-term memory we introduce an action relationships database reminiscent of the way symbols are associated in brain, and propose an adaptive mechanism for linking these actions with the different scenarios. The model then fills in the implicit context of the input and predicts relevant activities that could occur in the context based on a statistical action relationship database. The new data both of the more complete scenario and of the statistical relationships of the activities are reconsolidated into memory. Experiments show that our model improves upon the existing reasoning tool suggested by MIT Media lab, known as ConceptNet.

Keywords Memory reconsolidation · Natural language processing · Semantic network · Bayesian inference

K. Tu · D. G. Cooper · H. T. Siegelmann (✉)
BINDS laboratory, Department of Computer Science, University of Massachusetts Amherst, 140 Governor's Drive, Amherst, MA 01003, USA
e-mail: hava@cs.umass.edu

K. Tu
College of Computer Science and Engineering, South China University of Technology, 510640 Guangzhou, People's Republic of China

H. T. Siegelmann
Program of Evolutionary Dynamics, Harvard University, One Brattle Square, Suite 6, Cambridge, MA 02138-3758, USA

Introduction

One task of natural language processing (NLP) is to understand and represent the implicit meaning of a sentence. The implicit meaning can be obtained by analyzing the explicit meanings and information stored in memory. This may trigger a memory recall. Memory reconsolidation is the process where a recall of information leads to update or strengthen the already stored memory according newer existing information (Charniak 1991; Friedman 1998; Heckerman 1999). As semantic roles of verbs have been characterized with nouns (Kipper et al. 2006; Liu and Singh 2004a), they are reported to be used to predict the brain activity associated with the meanings of nouns by computational model on fMRI data (Loftus 1975). This suggests that the predicates help to understand the role of the associated entities. For example, an antelope chased by a tiger could be described with the following states of running away, being caught, get injured, etc., and this helps to understand the scenario of the antelope as a prey.

Inspired by this biological evidence where entities are internally understood by a set of actions, and by the evidence for modularity in brain memory system (Miller 1998) a scenario-adaptive memory system is built for inferring implicit meaning and prediction by combining the entering sentences with a Long-term memory Action Relationship Database (ARDB).

The basic data structure within our memory system builds upon semantic networks, introduced by Richens (Miller et al. 1990). These networks are useful for representing the semantic relationships between objects or concepts. They were shown to be powerful tools for knowledge representation and language processing, as well as logical description of events (Mitchell et al. 2008; Mizraji et al. 2009). Yet, their structure is rigid and less expressive than the options given in a natural language, so an extended form of semantic networks will be required.

Computers commonly make sentence understanding errors due to the multiple meanings of individual words. To avoid incorrect inferences, our system disambiguates words robustly by utilizing two knowledge bases WordNet (Kipper et al. 2006) and VerbNet (Liu and Singh 2004b). WordNet is an English lexical database that categorizes synonyms into groups, records word group relationships, and labels topic domains of some words; it can provide some word meanings for our system. VerbNet is another technology that describes constraints on the subject and the object of a verb and thus constitutes another way to disambiguate the word meaning. Input is matched with words in WordNet and VerbNet in order to find the unique word meanings and common topics; this correctly disambiguates the input words.

To obtain the implicit or full meaning of input sentences, we build a memory association system that is based on our Extended Semantic Network, described in "Information representation". We group the input words to topics and disambiguate them and then update the input via the Action Database (ARDB) from the long-term memory storage. We then update the scenario as well as the statistics in the ARDB in a reconsolidation-like process for future use.

We compare our new memory-inference model with the MIT Media Lab's ConceptNet (Murphy 2002; Niculescu-mizil 2007). ConceptNet is an open source knowledge base that is built in the semantic network structures. There are a given number of edge types such as "prerequisite of," "subset of," and "do;" the data can be thought of pairs of concepts connected with one of the allowed types of edges. For example ("is capable of" "cook-food" "be fun") is a pair with an edge where the name of the edge is written first. As this database is open to the public, many people have added information to it and created a large commonsense knowledge of pairs, presented in a semantic network format. ConceptNet has been suggested to be used to understand scenarios better than current natural language technologies due to the wide data it contains. While our focus is on Action Relations common-sense which is a subset of the information from ConceptNet we do propose particular ways to infer information in a Bayesian manner and this together with the topic-based disambiguation seems to provide a better tool for scenario understanding.

The following text is organized as follows. In "Information representation", we introduce the Extended Semantic Network (ESN). In "Working versus long term memory", we discuss how the working memory is constructed and updated by both input and the long-term memory. Section "The action relationship database" describes the action relationship database (ARDB) and the process of using it. Section "Demonstration" demonstrates our system on ball game scenarios and compares the results to those inferred by ConceptNet. We close in "Conclusion and predictions" with conclusions and predictions to be tested in future studies.

Information representation

A semantic network is a graph $G = (V, E)$, where V is the set of vertices representing the concepts or entities and E is the set of edges representing relations of the vertices. One application is the study of relations between words such as a lexical database of English and Gellish (a controlled natural language) models (Kipper et al. 2006).

We next propose an extended semantic network (ESN) structure to present information in both short-term and long-term memory.

Definition of ESN

Suppose $S = (s1, s2, \ldots, sn)$ is a set of sentences, an ESN is designed so that to store S as a weighted directed graph $G' = (V', E')$. Vertices $V' = \{vk|k = 1,2, \ldots, m\}$ are formed in an ESN for the subjects and objects in sentences (shown in Fig. 1), representing the entities of people or other things that exist in the described world. Edges constitute the directed relationships $E' = \{ed(Vi, Vj)|d = 1,2, \ldots, m\}$ between vertex Vi and Vj.

A simple English sentence has the grammar form: "subject + predicate". Some predicates can break into a form of "joiner + object", where the joiner contains a verb. This translates directly to two vertices and an edge in the ESN, see Fig. 1.

In our ESN we allow for a few additional edges over those in basic semantic networks.

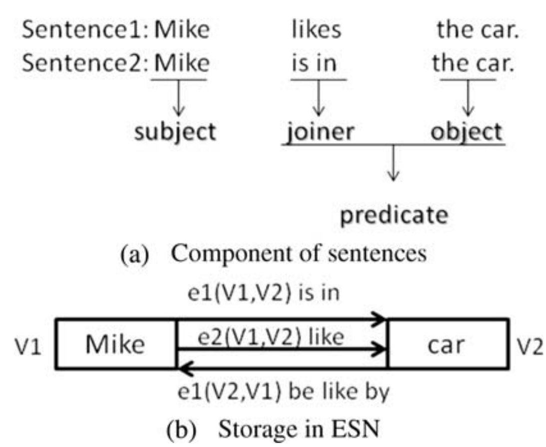


Fig. 1 **a** Two sentences and their structure; **b** storage in ESN. Entities of the subject and object are stored as vertices (denoted as *rectangle*), joiners are denoted as edges

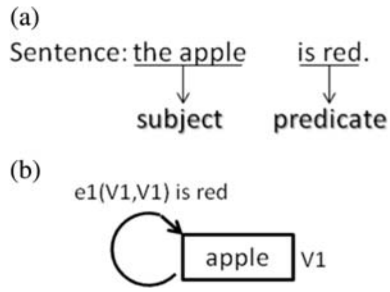


Fig. 2 A unary relation in an ESN. **a** A sentence with no object in its predicate, **b** representation of the sentence in an ESN

(a) Multiple relations between two entities cause multiple edges for a pair of vertices in the ESN. In Fig. 1, "is in" and "like" form two partial ordered relations between "Mike" and the "car" are stored as the edges e1(V1, V2) and e2(V1, V2).
(b) Another new edge is the reversed edge representing the passive voice of a verb such as the car "be liked by" Mike.
(c) Edges connecting two different vertices are regarded as binary relations. Loop edges are allowed in our ESN as unary relations where the predicates have no objects (e.g. "stops"), or when the predicates only describe a properties of their subjects (e.g. "is red" in Fig. 2b).

Updating the ESN

Edges are generated either (1) from input sentences as evidence or (2) from inferences based on the long-term memory. A probability is attached on the ESN's edges. The probability of an evidence relation is 1 while the probability of a non-evidence relation is based on the amount of belief this edge holds. A threshold governs the creation and deletion of new inferences: A relation will be removed from the working memory if its belief becomes lower than a given threshold. A new edge is added if its probability increases beyond the threshold.

Constraining inference via categorizing entities

"Mike" in Fig. 1 is an entity in the category of human, which can perform the action "like"; while "the car", classified as a vehicle, cannot "like" anything by itself. In order to ensure the relations are reasonable, the entities should be limited to perform several kinds of actions or to have certain properties according to their categories.

The constraints from VerbNet (Murphy 2002) are used to evaluate the validity of the meaning of vertices. For example, in the phrase "a bat catches insets," "catches" should have a human or animal as an agent. So the meaning of "bat" should be a "mammal" not a "club" for sports. If the evaluation indicates a mismatch between a vertex and an edge, the edge will be removed.

WordNet (Liu and Singh 2004a) can be used to hierarchically organize the vertices into categories or topics according to their meanings. A category can have subcategories, for example, "dog" in WordNet belongs to "canine," "canine" belongs to "carnivore," "placental," "mammal," and so on. In our model, entities will have properties and perform activities that fit their categories and parent categories, which are stored in our category tree.

Some words with multiple contexts in WordNet are classified into multiple categories. During inference, a word will be assigned to the category or a subcategory by matching the topic of the context in the ARDB and other information in the working memory. For example, the word "bank" has more than four meanings in its noun form and is related to topics such as "deposit", "depository financial institution", "flight maneuver", "slope" and so on. The sentence "he goes to the bank" can be confusing because the "bank" may related to any of the topics). The ARDB, which is part of the long-term memory, may contain statistical information needed to constrain the meaning of "bank". For instance the ARDB may have the following evidence: bank leads to (1) deposit 50 times, (2) jump in river 20 times, and (3) fishes in river 10 times. This would suggest that making a deposit is most likely although it is not necessary. In addition, other information in the working memory can add further evidence that the "bank" should be related to a "slope" (e.g. a vertex representing "river", has the same topic "water" as "sloping land beside water"). A sequence of input sentences that describes the same scenario may assist in revealing the meaning of "bank" when our system applies topic matching among the sentences.

Working versus long term memory

Working memory is the result of the interactions between the input, the long-term memory, and the current contents of working memory. Long-term memory consists of long-term ESNs and a statistical database of causal relationships between entities, the ARDB. This statistical database helps to determine the semantic relationship between words in working memory.

As words are input, they are matched first to the working memory. If a match is found, then the working memory is updated with the words. The words are also matched to the long-term memory. If a match is found, then the memory is retrieved and added to the working memory with additional associations. The working memory works on two levels simultaneously. The first level is the symbolic level, where the relationships between the words is expressed in a semantic network, and the second is the sub-symbolic level, where the underlying probabilities can be used to activate additional parts of the network in order to predict new related words.

The symbolic structure allows us to connect with the action common sense knowledge bases, such as VerbNet, to benefit from the context, and the sub-symbolic structure allows us to ignore the labels and to just look at the connectivity of the graph: the probabilities on the graph are a way to model spreading activation of semantics (Riccio et al. 2006), while thresholds constrain the process. When an activity is observed for an entity and the activity matches one in long-term memory, the activity is marked as retrieved, and later can be updated to better match the working memory in an adaptive reconsolidation like process.

The vertices state for ESN update

A vertex (representing an entity in the memory) in the extended semantic net has three kinds of states: (1) recalled working memory (RWM), (2) novel working memory (NWM) and (3) inactive. Table 1 shows the differences among the three states of vertices. An RWM vertex represents an entity for inference. A copy of its information will be stored in the working memory and its renewed relations with other vertices will be updated to the long-term memory. An NWM vertex will become an RWM vertex if it is linked to another RWM vertex in the reconsolidation like process with a high probability relation. An inactive vertex is not relevant for inference.

The translated sensory input, such as text, is first composed into working memory as NWM. Then the long-term memory is scanned for matches to the working memory. NWMs that find a match to nodes in long-term memory are converted to RWMs, and the associated nodes in long-term memory are copied to working memory as RWMs as well.

The action relationship database

The Action Relationship Database (ARDB) uses Bayesian statistics to update its knowledge as more input examples arrive and thus can learn the strength of the connections as well as new ones. Technically speaking it is embedded in a structure called Bayesian network (Richens 1958; Sara 2000; Sowa 1974, 1987). The ARDB can be presented graphically but it differs from the semantic and the extended semantic networks. In the ARDB, each vertex represents a random variable and has an attached conditional probability table that states how likely the variable is to occur given the occurrence of the parent vertices. The graph can also be understood as the ''belief about actions'' that the person created out of the many input sentences it saw. Figure 3 illustrates how the predicate "be attacked by tiger" uses the ARDB to predict the activities of an antelope.

The general process of matching input with actions starts as described above. When an edge from the ESN in working memory matches a node in the ARDB, the edge is treated as evidence that the statement of the node is true. Figure 3b shows the part of the ARDB that is relevant for inference on the edge-node relationship in the ESN 'be attacked by'-'tiger.' From this evidence, the system would infer that the antelope was injured. This would then be added to working memory as an inferred part of the ESN.

The actions from the ARDB are broken into the form "edge (+vertex)" as they are translated into the ESN structure. This enables a bi-directional information flow between the ARDB and the ESN. The information of a subject's activities are propagated from the ARDB to the working memory by a similar translation. In the case of the antelope, there would be two edges added: (1)'the antelope'-'was injured by' 'the tiger' and (2) 'the tiger' 'injured' 'the antelope.' The system performs all edge operations (such as search, add or remove) in working memory.

Inferring from actions

There main steps to infer new information from our activity-based model are:

- Activate related vertices and move them to the working memory. New entities from arriving sentences are automatically set as active in the working memory.
- Classify the entities into categories.

Table 1 The three states of vertices in the memory have the different uses shown here

State	In long-term memory	In working memory	For inference	Information update	Assigned a category
RWM	Yes	Yes	Yes	Yes	Yes
NWM	No	Yes	Yes	No	No
Inactive	Yes	No	No	No	N/A

"For inference" refers to filling in missing data, "information update" refers to the LTM being updated based on changes to RWM nodes, and "assigned a category" refers to the topic matching performed with the common sense databases

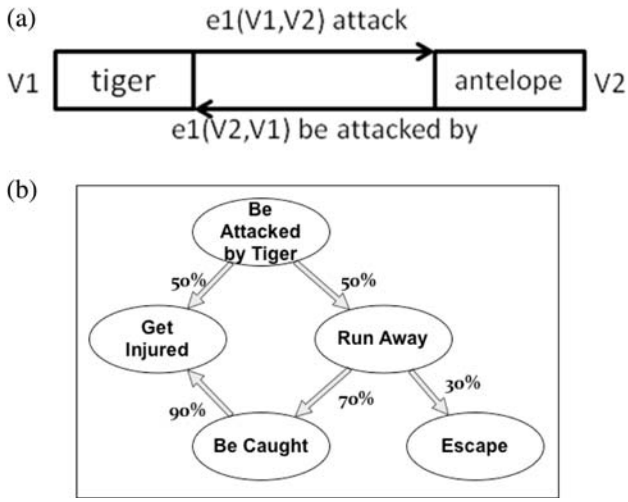


Fig. 3 Predicates denoted as nodes in an Action Database for a scenario about an "*antelope*". **a** Initial information in working memory; **b** related nodes (denoted as *ovals*) from the Action Database

- For a entity in the working memory, the syntax constraints from VerbNet are applied to select relevant actions from the ARDB that may be affected by the evidence.
- Further constraining the relevant actions from the ARDB based on the topics via link paths.
- Combine the actions that remained after constraining the ARDB into one on-line relevant ARDB. Update the joint conditional distribution of the shared nodes.
- Expanding the relevant actions by searching for dependency-connecting paths from the evidence node to infer all related activities from the on-line ARDB.
- Calculate the posterior probability of the variables in the on-line ARDB based on the input evidence.
- Update the working memory and the long-term memory accordingly.

Demonstration

To test the effectiveness of our memory model, we show how to use it to infer a whole scenario from input sentences. We will also demonstrate that our inference constitutes an improvement over the reasoning of the MIT media lab's ConceptNet.

Information processing

Suppose the first sentence that comes as an input is: "Mike swings a bat". In the ConceptNet database, there are many pairs that may be associated with this sentence: (CapableOf "baseball player" "swing bat"), (SubeventOf "play ball" "swing bat"), (Isa "bat" "mammal"). Each relationship is given a number of how many times (frequency) it was supported by users.

We create our Action Relationship Database (ARDB) from ConceptNet and initialize the probabilities. The threshold of probability for adding an edge in memory will be fixed to 0.65. Our model first stores the sentences in the long-term (Fig. 4a) and working memory as evidence.

The second step is to classify entities into categories. "Mike" belongs to the category of human. The "bat" has several contexts as a noun: (1) a turn to get hit in baseball, (2) a mammal (3) a club for a ball game and (4) squash racquet (Fig. 4b). These definitions are stored as NWM vertices. As a result, actions related to these meanings can be selected for inference.

The third step is to select all relevant topics. While ConceptNet would prefer the ball game topic over the mammal topic, based on higher frequency by users, our system would still leave the two topics available in case a following input sentence might associate with the less likely topic. "Swing bat" is thus related to a ball game, and the system keeps the the less likely interpretation that the "bat" means a kind of mammal and "Mike" has caught it to play with.

ConceptNet decided that Mike is engaged in a ball game without using details of the topic and deduce that "Mike is playing tennis or baseball".

To continue with our example, we suppose that the next sentence is "John throws a baseball." The "baseball" has the topic of a ball game. Thus, actions from the ARDB related to baseball outweigh the others and are selected. Figure 5 shows the data related to the predicates that are inferred from ConceptNet.

ConceptNet will incorrectly deduce that "Mike" "plays tennis" using the relation of "play tennis" and "hit ball", because it does not update it's probabilities based on incoming data, rather it uses the prior probabilities based on frequencies of occurrence in the past.

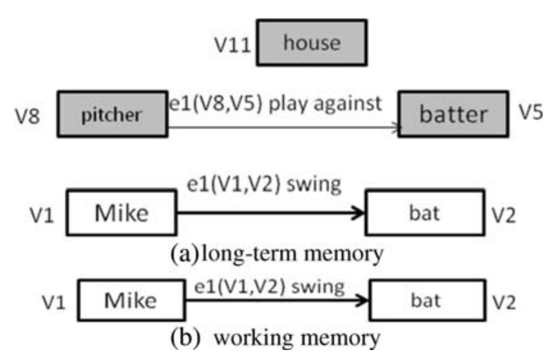


Fig. 4 **a** "*Mike swings a bat*" stored in long-term memory. *Grey rectangles* are inactive vertices, "*Mike*" and "*bat*" are RWM vertices. **b** Pre-process of information in working memory for inference

In our model, "play tennis" is removed, since it does not fit the chosen topic of baseball. The relevant on-line Action Database, a subgraph of the ARDB, used for these input sentences is as follows in Fig. 6b.

Figure 6 shows that the main topic in the working memory is decided according to the RWM vertices, and a selected set of actions from the ConceptNet database are chosen with similar topic keywords to constitute the on-line ARDB.

The next step is to find dependency connecting paths (see Tulving and Thomson 1973) for the Action nodes related to the evidence "swing bat" and calculate probabilities. We see that "is baseball pitcher" is independent of the evidence and the probabilities are calculated with the distributions as in Table 2.

"Throw ball" has a probability lower than the threshold and will not be added. Other nodes with high probability are added to the working memory. "ball," "batter," and "baseball player" are denoted as NWM vertices. If the probabilities of their edges are still beyond threshold after inference for all the other RWM vertices, the NWM vertices will be changed to RWM vertices. Figure 7 shows the nodes added to Mike in working memory.

A similar inference of "John's" activities is made. The assertion (SubeventOf "play football" "throw ball") from ConceptNet does not match the keywords in the topic of our current working memory and thus is eliminated.

Edges are added to the working memory according to their probabilities in Table 3. The inference is now over. The NWM vertices linked to "Mike" and "John" with high probability edges are changed to RWM vertices and updated to the long-term memory.

From the input sentences and the fact they came from the same scenario, we inferred new information for "Mike" and "John." This information can be easily output as meaningful sentences shown in Table 4.

Our model improves on the inference of ConceptNet and avoids the false conclusion "John play football" that was based on "throw ball" without consulting the topic of "baseball player". A connection between the sentences of the scenario is being made in our system: In Fig. 4, there are two inactive vertices "pitcher" and "batter". Batter and pitcher are subcategories of baseball player. The relation "is" suggest that "Mike" belongs to "batter" and "John" belongs to "pitcher".

Once these relationships are inferred, the pitcher and batter nodes become active vertices (Figs. 8, 9) and their relation "play against" is recalled. With the updated information: "Mike is a batter" and "John is a pitcher", the recall can suggest "John" is playing against "Mike".

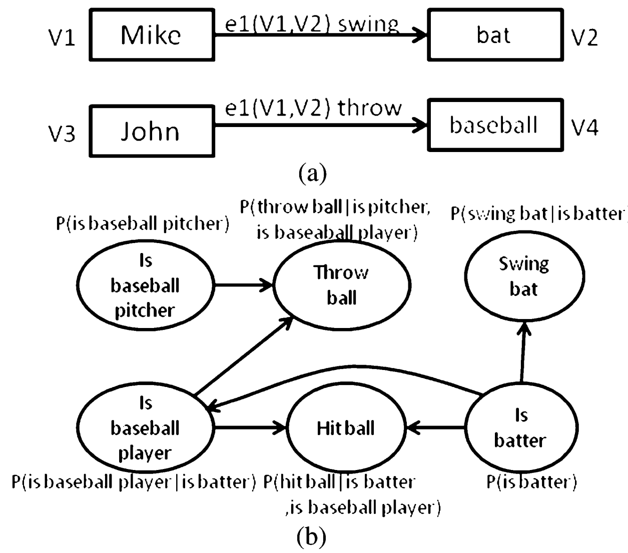
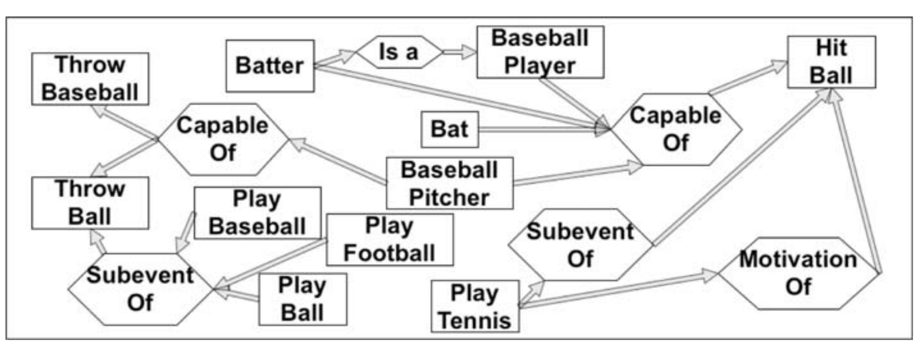


Fig. 6 Memory update for "*Mike*" **a** new information added to working memory, **b** an on-line Action Database for activities of "*Mike*"

Table 2 Probability of activities of "Mike"

Predicate for "Mike"	Probability	Value
Is batter	P(Is batter)	0.77
Hit ball	P(hit ball	0.67
Is baseball player	P(is baseball player)	0.86
Throw ball	P(throw ball)	0.62
Is baseball pitcher	P(is baseball pitcher)	N/A

Fig. 5 The relationships that are inferred by ConceptNet given the sentences "Mike swings a bat." and "John swings a baseball."



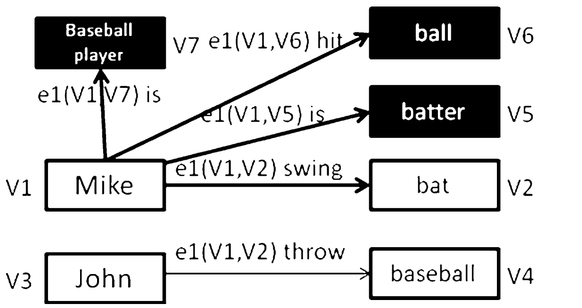


Fig. 7 Working memory update for "*Mike*". NWM vertices are denoted as *black rectangle*

Table 3 Probabilities of predicates for "John"

Predicate for "John"	Probability	Value
Is batter	P(Is batter)	0.60
Hit ball	P(hit ball)	0.61
Is baseball player	P(is baseball player)	0.84
Swing bat	P(swing bat)	0.50
Pitch	P(pitch)	0.7
Is baseball pitcher	P(is baseball pitcher)	0.65

Table 4 Newly update memory

Entities	New information
Mike	"Mike hits the ball"
	"Mike is a batter"
	"Mike is baseball player"
John	"John is a pitcher"
	"John is a baseball player"
	"John pitches".

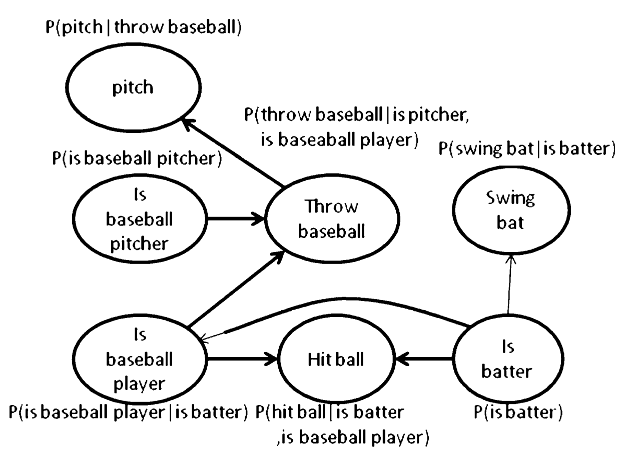


Fig. 8 Memory update for "*John*": on-line ARDB constructions

Memory reconsolidation

In changes to memory—we need both to store the expanded scenario and update the ARDB. Once the relationships

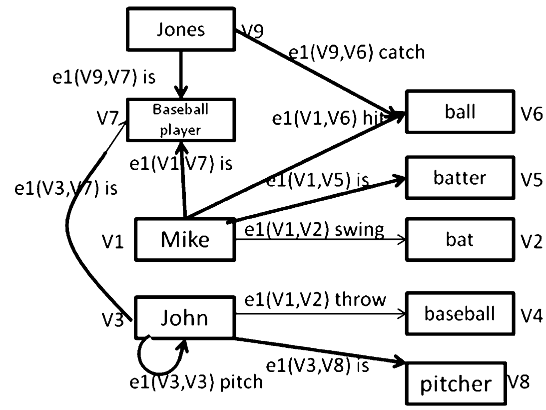


Fig. 9 Memory update for "*John*": updated working memory

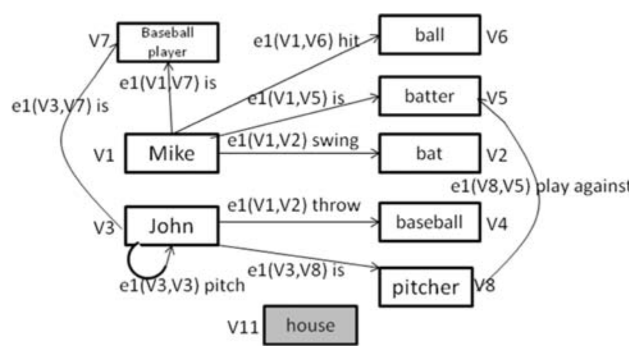


Fig. 10 Long-term memory ESN after update

between the entities via the actions are inferred, the ESN is updated to reflect the change as shown in Fig. 10 and this is also transferred to long-term memory. In addition the ARDB is updated by considering the new scenario and modifying the statistics underlying the database so that with an increasing amount of information, the ARDB will have a different world model.

This is analogous to Memory Reconsolidation and adaptive generalizations in the human brain, where memories that are recalled can be modified by novel or different experience after retrieval and where adaptive understanding is then created.

Conclusion and predictions

The work is based on the assumption that the actions related to entities constitute a chief source for understanding and inference. Based on this assumption we proposed a particular memory model that infers about scenarios via commonsense that is given in the form of an Action Relation Database (ARDB).

The model we propose follows the notion of reconsolidation. When memory is recalled from long-term memory this memory becomes weak and can change and update. Such reconsolidation is a strong tool to assure flexibility in understanding of dynamic environments.

We in particular propose a memory reconsolidation model to represent and infer implicit meanings and fuller scenarios from sentences in the term of natural language together with inferring relations between actions from these scenarios.

As part of the new memory model, we introduce two new data structures. One is the Extended Semantic Network (ESN) that can better represent natural language relationships between entities by allowing parallel directed edges as well as self loops. The second is the Action Relation Database (ARDB) that is constrained and recreated on-line in steps to fit the scenario in the working memory.

Basing on the particular memory model and the algorithms for inference used within we suggest that the use of commonsense is done in terms of actions, and that this process occurs in steps: first a more general set of actions is chosen and then it is refined to fit the model. Such phenomena can be tested by both fMRI as well as psychophysics experiments. Our second prediction has to do with the algorithmic view of the process of reconsolidation. We propose that reconsolidation is relevant for both the immediate memory, which is the updated scenario, as well to a generalization based on Action Relations. The latter has not yet been demonstrated in humans or lower animals.

In terms of the developed technology our tool is proven stronger than current inference engines on scenarios based on natural languages, and it even seems to outperform the known MIT Media Lab's commonsense database called ConceptNet. The power of our technology is in three different directions. First it demonstrates how to constrain false deduction by the use of topic recognition and word disambiguation before finalizing the relevant ARDB. This saves common false deductions occurring in ConceptNet. Secondly, our model demonstrates how to propagate probabilities and beliefs farther away in the relevant ARDB and thus can reach further conclusions from actions. Third our model connects sentences of the same scenario within the single working memory and it thus better combines sentences and their topics into one meaningful scenario.

Acknowledgments This work is supported by the ONR grant N000140910069 and partly by Chinese Scholarship Council.

References

Charniak E (1991) Bayesian networks without tears. AI Magazine 12:50–63

Friedman N (1998) The Bayesian structural em algorithm. In: Mora GFCaS (ed) Uncertainty in artificial intelligence: Proceedings of the fourteenth conference. Morgan Kaufmann, Madison, Wisconsin, pp 129–138

Heckerman D (1999) A tutorial on learning with Bayesian networks. In: Jordan M (ed) Learning in graphical models. MIT Press, Cambridge, MA

Kipper K, Korhonen A, Ryant N, Palmer M (2006) Extending VerbNet with novel verb classes. In: Proceedings of the 5th international conference on language resources and evaluation. Genoa, Italy

Liu H, Singh P (2004a) ConceptNet: a practical commonsense reasoning toolkit. BT Technol J 22(4):211–226, Kluwer Academic Publishers

Liu H, Singh P (2004b) Commonsense reasoning in and over natural language. In: Proceedings of the 8th international conference on knowledge-based intelligent information & engineering systems (KES'2004). Wellington, New Zealand, September 22–24. Lecture notes in artificial intelligence, Springer

Loftus EF (1975) A spreading-activation theory of semantic processing. Psychol Rev 82(6):407–428

Miller G (1998) Nouns in WordNet. The MIT Press, USA

Miller G, Beckwith R, Fellbaum C, Gross D, Miller KJ (1990) Introduction to WordNet: an on-line lexical database. Int J Lexicography 3:235–244

Mitchell TM, Shinkareva SV, Carlson A, Chang K-M, Malave VL, Mason RA, Just MA (2008) Predicting human brain activity associated with the meanings of nouns. Science 320:1191–1195

Mizraji E, Pomi A, Valle-Lisboa JC (2009) Dynamic searching in the brain. Cogn Neurodyn. doi:10.007/s11571-009-9084-2

Murphy KP (2002) Dynamic bayesian networks: representation, inference and learning. University of California, Berkeley, p 268

Niculescu-mizil A (2007) Inductive transfer for Bayesian network structure learning. In: Shen MMaX (ed) Proceedings of the 11th international conference on AI and statistics (AISTATS), vol. 2, pp 339–346

Riccio DC, Millin PM, Bogart AR (2006) Reconsolidation: a brief history, a retrieval view, and some recent issues. Learn Mem 13:536–544

Richens RH (1958) Interlingual machine translation. Comput J 1:144–147

Sara JS (2000) Retrieval and reconsolidation: toward a neurobiology of remembering. Learn Mem 7:73–84

Sowa JF (1974) Conceptual graphs for a data base interface. IBM J Res Dev 20:336–357

Sowa JF (1987) Semantic networks. Encycl Artif Intell

Tulving E, Thomson DM (1973) Encoding specificity and retrieval processes in episodic memory. Psychol Rev 80:352–373