JAM: Java-based Associative Memory

Robert Pröpper, Felix Putze, Tanja Schultz

Abstract In dynamic environments, conversational dialog systems have to deal with ambiguous input, topic shifts and the users' limited memory resources. Therefore, systems need to model cognitive processes of its users to predict "what is on the user's mind". In this paper, we introduce JAM, a cognitive model of associative memory designed for the application in dialog systems. JAM is able to estimate dynamic processes like association, concept drifts and forgetting of information. We describe the data structures and algorithms developed to support these operations and present evaluation results, including the outcome of a survey conducted to compare the results of JAM to human associations.

1 Introduction

While spoken dialog systems have matured to a point where they are routinely employed in static and controllable scenarios, such as virtual call-center agents, they still lack flexibility and robustness in dynamic scenarios. Examples for such applications are human-robot interaction, in-car systems or portable companion technology. In such scenarios, a major problem is the fact that it is hard to estimate "what is on the user's mind". The conversation may shift slowly from one topic to another due to evoked associations in the user's mind. External stimuli may cause sudden changes of focus, while other discourse items may fade out and eventually be forgotten by the user. These effects become particularly important when the interaction becomes less task-driven and more conversational, as envisoned for many natural interaction systems. Another challenge in verbal human-computer interaction is the ambiguity of natural language. While humans are able to resolve it by referring to a shared context, computer systems mostly lack this ability. Providing knowledge about hu-

Cognitive Systems Lab (CSL)
Karlsruhe Institute of Technology (KIT), Germany
e-mail: {felix.putze, tanja.schultz}@kit.edu, {robert.proepper}@student.kit.edu

man association mechanisms is one step towards enabeling systems to understand the underlying processes.

In this paper, we introduce JAM, a Java-based associative memory framework, which provides a way to model dynamic association processes of the user's memory. JAM is able to determine the most likely associations of a human for a given memory configuration and a set of new stimuli. It provides this information to a dialog system or other speech processing components such that these systmes can resolve ambiguities or determine the user concerns. The flow and organization of those processes and their consequences are studied by cognitive psychologists. For example, there is the phenomenon called the "Moses Illusion" [3]: Participants of a study were asked questions like "How many animals of each kind did Moses take on the Ark?". Even though most christians know that it was Noah and not Moses who built the ark, many give the answer "two", which is wrong in terms of deductive reasoning but plausible in terms of association. Observations like this underline the importance of using a cognitively sound and validated model of memory.

Since the content and context of an interaction change over time, modeling those dynamics is a crucial part of JAM: Different associations are formed as new items come into focus, integrated with previously active items while old items are gradually forgotten.

Possible applications for JAM include speech processing systems and translation systems to enhance them by helping to resolve ambiguities - a result obtained with purely statistical methods is more likely to be correct if it is also part of the associative context in JAM. As a user modeling component in a dialog system, it may identify topics which are most relevant to the user in the current context without an explicit request. Another very important domain of JAM is user simulation to automatically create training and evaluation scenarios for dialog systems: Here, the model can be used in a generative fashion to predict plausible associations for a situation and derive consistent speech acts and utterances of the simulated user.

2 Related Work

There are a number of systems and studies which acknowledge that the the user's memory is relevant for the design of dialog systems. For example, it is accepted that the design of a system has to accout for the limitations of working memory. Koedinger et al. [8] deal with cognitive tutoring systems which employ strategies for reducing memory load by removing irrelevant information or by visualizing the discourse structure. Wolters et al. [18] investigate the influence of different strategies for information presentation on working memory and compare the trade-off between shorter utterances at the cost of more complex discourse structures. Jameson et al. [6] model in a Bayesian network several factors of human cognition which have an impact on dialog system performance, including memory capacity limitations. However, we are not aware of many systems which explicitly model how activation of memory items develops, spreads and diminishes: Most state-of-the-art dialog systems acknowledge the difference between the discourse model of the system and the set of beliefs of the user. For example, many systems have a notation of grounding to model presence or absence of a common ground and can thus model potential discrepancies between the system's perspective and the state of the user's mind [12]. However, once information is assumed to be correctly processed, most systems cannot handle the user's dynamic memory processes, i.e. they do not cover the activation of new concepts by association or their removal by forgetting or concept drift. Lieberman et al. [9] used a large associative database as an additional information source for an automatic speech recognizer and showed how incorporating knowledge of human associations can improve the results of statistical models. However, this approach does not explicitly model cognitive processes.

In a survey on statistical user simulation, Schatzmann et al. [16] list the modeling of concept drift as one of the major future challenges, i.e. "the problem that user attributes change over the course of interaction with a system". The authors point out that current user simulation approaches for dialog system training assume stable goals during the course of a dialog. As their examples show, the user's goals are heavily influenced by the information which is currently actively processed. A modeling of concept drift for dialog systems therefore requires a dynamic model of this information. Pietquin [13] proposes a simple probabilistic model of the user's memory as a component of a user simulation framework. Memory is updated after every dialog turn and is modeled as a probability distribution conditioned on the previous memory state and the system utterance as perceived by the user. The hidden agenda model by Schatzmann and Young [15] introduces a variable user agenda, depending on the user's goal and the course of the interaction. The agenda is maintained as a stack which can be manipulated by push and pop operations. This process is described in a probabilistic model and used to generate plausible and consistent user behavior for the training of dialog strategies. The virtual human proposed by Kenny et al. [7] is able to generate realistic human behavior in a simulation environment with the help of the cognitive architecture SOAR. SOAR includes models of short and long term memory and simulates human behavior which follows theories on cognition.

In earlier work [14], we showed the feasibility of simulating interactions between a human car driver and a virtual co-driver using a predecessor of JAM in a user study. In [14], human judges rated interactions which consisted of utterances generated based on a memory model as similar to handcrafted interactions. In the current paper, we tackle the challenge of associations and concept drifts by proposing a method to systematically model the dynamics of the user's memory and its associative processes. We use established cognitive models as a starting point and adopt them to fit the needs of the domain. The described static algorithms and much of the knowledge representation in JAM were inspired by LTM^{C} from Schultheis, Barkowsky and Bertel [17]. LTM^{C} was developed as an enhanced long-term memory for the cognitive architecture ACT-R. The ACT-R theory was developed by Anderson [1] as an integral model of the human mind. It introduces the concept of spreading activation for memory items which is used in a similar way in both LTM^{C} and JAM. To fill the knowledge base of JAM, we access different large common sense databases. The ConceptNet database is part of the Open Mind Common Sense (OMCS) project started at the MIT Media Lab [4]. The data is compiled from common sense statements entered by many users at the OMCS website. Cyc is a commercial project by Cycorp which combines a large general knowledge database with a reasoning engine [11]. The data is entered by employees of Cycorp. OpenCyc contains a freely available subset of the Cyc database.

3 Architecture

To describe the structure of the knowledge representation we are working on, we will use the terms *concept* and *association*. A concept is an object of common sense knowledge. It could be a physical object, an attribute, an activity or an abstract idea. Associations are links between concepts. An example for an association is the statement “The KIT is located in Karlsruhe”, where “KIT” and “Karlsruhe” are concepts and “is located in” is the association between them.

3.1 Knowledge Structure

For the most part, we have adopted the graph-based knowledge representation of LTM^C described in [17]. There are two basic types of nodes in the knowledge graph: concept nodes and association nodes. Edges generally have no other meaning than describing a general relationship between two nodes. The statement “The KIT is located in Karlsruhe” would be encoded in three nodes as shown in figure 1.

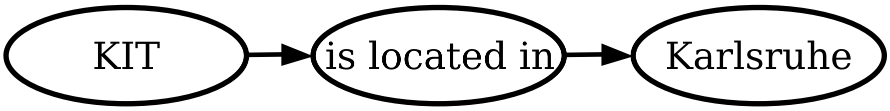


Fig. 1 An association in JAM. Two concepts and an associations of the type “is located in”

In most cases, the ConceptNet data fits better with the associative usage scenario in JAM. This is not surprising since in the original method of data collection by OMCS, users were asked about their associations (see [10]). In addition, OpenCyc is mainly an ontology - there are few associations that are not of the type “is a”. While many relations in the OpenCyc corpus contain metadata (e.g. “Wn_20_synset_Germany_noun_1”) which cannot be easily mapped to a concept or association, the ConceptNet data also contains subjective associations (e.g. “Germany has good beer”) which may be of great interest. However, there are areas of knowledge (e.g. specific people) not covered by ConceptNet and for which Open-

Cyc is a better choice. Currently, data can be imported from both OpenCyc or ConceptNet.

3.2 Memory Dynamics

Each node in JAM has an activation value. The activation is the likelihood for a node to become part of working memory and receive attentional focus. It also serves as an indicator for the amount of time it takes for the node to be retrieved from long-term memory. A node can get activated in two ways: By an external stimulus or through spreading activation, i.e. the propagation of activation from activated nodes to associatively linked nodes.

A typical example for an external stimulus is hearing about an item in a conversation. However, a stimulus does not always have to be verbal. An object coming into view could provide a stimulus. Also, previous knowledge or events can influence node activation. When n items get stimulated simultaneously, each node receives $\frac{1}{n}$ of the activation a single node would receive.

Spreading activation is the main mechanism used in JAM to trigger associations. It is implemented as a depth first traversal of the graph starting with the set of stimulated nodes: each node n spreads part of its activation to all nodes linked to n, which in turn spread part of their received activation. The equation for the amount of activation a node n receives from its predecessors is:

$$\mathrm{SpreadReceived}(n, n_{\mathrm{pred}}) = \frac{\mathrm{TotalSpreadReceived}(n_{\mathrm{pred}}) * f_{\mathrm{dampening}}}{\mathrm{NumberOfNeighbors}(\mathrm{Predecessor}(n))} \tag{1}$$

$$\mathrm{TotalSpreadReceived}(n) = \sum_{n_{\mathrm{pred}} \in \mathrm{Predecessors(n)}} \mathrm{SpreadReceived}(n, n_{\mathrm{pred}}) \tag{2}$$

Spreading stops once the amount of activation to be spread from a node falls below a threshold. This is necessary to keep the model computable, but it also follows the all-or-nothing principle in the human nervous system: neurons only transmit a signal if their received signal strength is above a certain threshold.

The free parameter $f_{\mathrm{dampening}}$ can be used to restrain the activation spreading. It can be thought of as a measure of creativity in free association - higher values result in more associations with less direct links to the stimulated input. The amount of activation a node spreads is reciprocal to the number of its neighbors to model the fan effect [2], which describes the strengh of associations.

Over the course of a conversation, new items will be stimulated while the activation of old items will fade. In order to keep activation values realistic over time, we introduce an activation decay mechanism. We use the entire activation history of a node (resulting from stimulation and spreading) to calculate its activation at a given time. The total activation of a node n at the time t_{current} is given by the equation:

$$\begin{aligned}\text{Activation}_n(t_{\text{current}}) = &\textstyle\sum_{t \in \text{History}_n} (f(t_{\text{current}} - t) * \text{ActivationHistory}_n(t)) \\ &+ \text{SpreadReceived}(n)\end{aligned} \tag{3}$$

$\text{ActivationHistory}_n(t)$ returns the activation value of the node n at the time t. This equation is not recursive - the activation history contains only values directly resulting from spreading. $f(x)$ is defined as:

$$\begin{aligned} f(x) &= \;1, \quad \text{for } x < 0 \\ f(x) &= \tfrac{1}{x+1}, \text{ for } x \geq 0 \end{aligned} \tag{4}$$

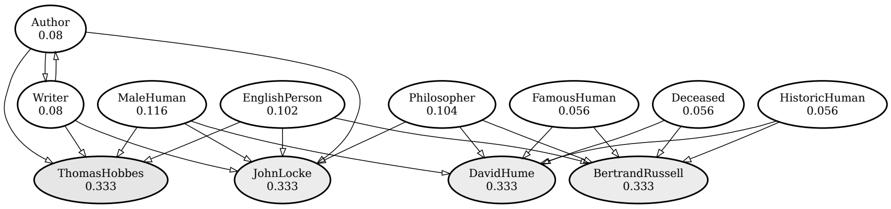
This function decreases almost linearly for small values of x (i.e. t is close to t_{current}) and then asymptotically approaches 0. Because it is multiplied with the activation history, this means that the total activation of items which are not recently stimulated will drop fast, ensuring that newly stimulated items have a higher activation. Items that have not been stimulated for a while will have a very small activation but are still distinguishable from items that have never been active. Figure 2 shows an example of how activation evolves over three spreading iterations.

4 Implementation

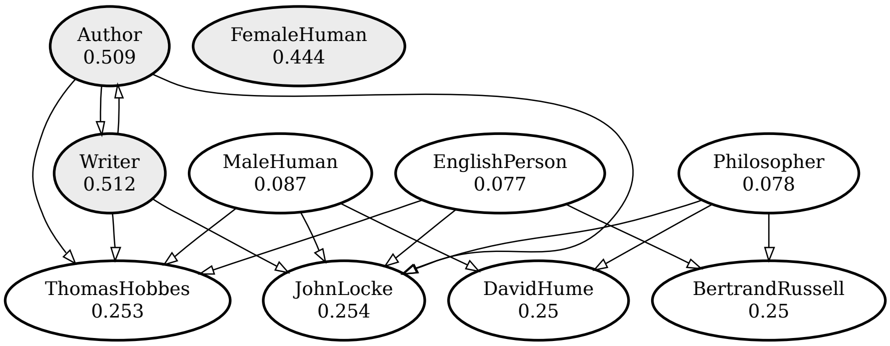
JAM is implemented as a Java library, so it can be used by Java applications as well as a standalone server accessible remotely by any application, e.g. a speech processing system or a dialog manager. It is designed to accommodate a variety of different implementations for its core concepts. The central interfaces are `Network` and `Session`.

A `Network` object represents our knowledge graph. There are currently importers for OpenCyc and ConceptNet, but a `Network` can also be manipulated directly through the API, e.g. to create a customized, application-specific graph. Two different implementations of `Network` can be used: For platforms with sufficient main memory (around 1 Gigabyte for OpenCyc or ConceptNet), `POJONetwork` provides a fast implementation where all data is stored as Java objects in memory. With `HGDBNetwork`, a slower implementation with a much smaller memory footprint based on the HypergraphDB [5] database is also available.

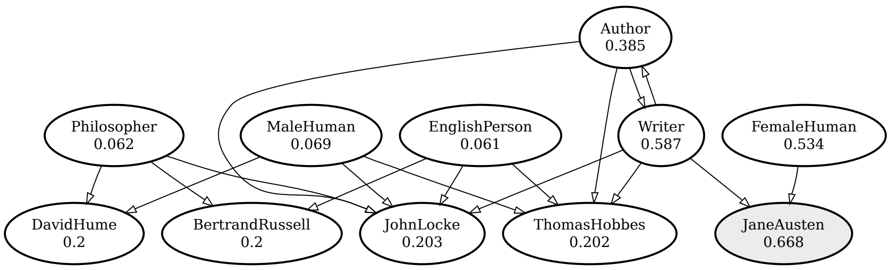
The `Session` interface provides a mutable view on a `Network` but does not change it. It mainly contains activation histories for all nodes. Multiple `Session` objects can operate on the same `Network` object simultaneously, e.g. to maintain different hypotheses of the memory state. An implementation of this interface also contains the dynamic processes that operate on the knowledge stored in the knowledge graph. For example, the algorithm we have described in section 3.2 is contained



(a) Stimuli “BertrandRussell”, “DavidHume”, “JohnLocke” and “ThomasHobbes”



(b) New stimuli “Author”, “Writer” and “FemaleHuman”



(c) New stimulus “JaneAusten”

Fig. 2 Results of consecutive stimulations with spreading activation. This could be part of a conversation about English authors. At first, four specific people were stimulated (colored gray). Next, three categories were stimulated, one of which (“FemaleHuman”) did not fit for any of the previous stimuli. The final stimulus is another specific person that is part of the new category. The effects of introducing new activation through stimulation is counteracted by the decay of the activation history. In this example, OpenCyc was used. All associations are of the type “IsA”.

in an implementation of `Session` named `SessionSpread`. Other `Session` implementations are also available.

5 Evaluation

JAM is supposed to work as a model of the human associative process. If JAM performs optimally, it should provide associations identical to human associations. Therefore, we compare in our evaluation the JAM results to human associations. The evaluation is divided into two parts: First, we have conducted a survey to investigate the results of a single spreading process. Second, we simulated a conversation of two instances of JAM to qualitatively evaluate their behavior over time. The ConceptNet database was used in all evaluations. The entire database was loaded, leading to 320000 nodes and 480000 links. ConceptNet was chosen over OpenCyc because of its closer relation to common human associations, which is what we are evaluating here.

5.1 Survey

To compare the associations made by JAM using the ConceptNet database with those of humans, we developed a questionnaire and asked 20 people to fill it out. All participants were students or employees of the KIT between 20 and 30 years of age and all participated in the same week. None of them were English native speakers. The questionnaire included five sets of three related stimuli and participants were asked to write down their first two associations for each set. We then activated the same sets of concepts in JAM in order to compare the results with the answers of the subjects. The presented stimuli were:
a) go restaurant, fork, diminish own hunger
b) tennis, soccer, volleyball
c) germany, france, spain
d) hamster, dog, cat
e) pen, work, desk

To evaluate if our spreading activation implementation using the ConceptNet data returns plausible results, we checked if the most frequent answers were reflected in the nodes with the highest activation. The results are listed in Table 1. It shows in the first column the different sets of stimuli and in the second column all associations that were given by more than one person, ordered by frequency. For each association, it also contains the number of times it was stated. Column three contains its activation rank in the results of JAM ("-" indicates that the concept was not activated at all or that its activation was negligible). If the JAM output contained a very similar concept with a higher rank, we have also included it.

Table 1 Survey Results. The names of stimuli and associations are shown as they appear in the ConceptNet database.

Stimuli	Association (# people)	JAM Rank
go restaurant fork diminish own hunger	eat (7)	1
	food (6)	6 (eat food 2)
	plate (3)	4
	hunger (2)	- (hungry 10)
tennis soccer volleyball	sport (9)	1
	ball (5)	15 (ball sport 5)
	basketball (3)	-
	team (2)	- (team sport 4)
	television (2)	-
	play (2)	- (play volleyball 8)
germany france spain	europe (14)	2
	country (8)	1
	italy (5)	-
	language (3)	-
	greece (2)	-
	holiday (2)	-
hamster dog cat	pet (12)	1
	animal (5)	4
	rabbit (2)	-
	mouse (2)	- (rat 2)
pen work desk	write (6)	1
	office (5)	3
	paper (5)	2
	university (4)	-
	computer (3)	6
	school (2)	10
	chaos (2)	-
	money (2)	-

Overall, the results are very encouraging. Each association shared by at least a third of the participants was also highly activated in JAM. In most cases, the first items on both lists were the same. However, there are a few interesting observations: Some of the associations of the participants did not fit with all of the stimuli - they seem to be associated with only one or two of the stimuli items. This behavior is also reflected by the JAM output (e.g. “team” / ”team sport” does not fit very well for “tennis”).

The vast majority of associations mentioned by participants were only one word while JAM concepts sometimes include multiple words (this is especially visible with the *sports* stimulus). When presented a stimulus consisting of multiple concepts of the same type, the participants often associated another example of the type (like “basketball” for *sports* or “italy” for *countries*) - these types of associations do not seem to be represented very well with JAM (even though the restriction of spreading to hierarchical siblings was not activated).

There are some external factors that can influence a study like this. Among the most important ones is language. Since we used the English version of the ConceptNet database (there are versions in other languages available, but they are much smaller), we conducted the survey in English. However, none of the participants were English native speakers. Depending on the individual language proficiency of the participants, this could affect the results - especially if one has to translate the stimuli, associate in one's native language and then translate the associations.

Another consideration is the order of the items. The order of the sets and inside of the sets depicted in Table 1 is the same as on the questionnaire. One participant had the association "food" for the *pets* stimuli, which could be attributed to priming by the earlier *eating* related stimuli. The order in which the three stimuli of each set were given could influence the resulting associations as well. The order in all questionnaires was the same. The JAM queries did not incorporate order considerations: Each set of stimuli was given synchronous and without previous activation from other sets because the dynamic aspects of `SessionSpread` have a much stronger impact on the results than the more subtle effects described here.

5.2 Conversation

In order to evaluate the dynamic behavior of JAM, we test it in an evolving context. We decided to simulate a "conversation" since this is a common example for a possible usage of JAM. We use two instances of JAM which communicate with each other to test if the course of the conversation remains realistic when all associations in a dialog are generated by JAM.

Initially, we stimulate both instances with the same concepts. We then randomly choose one activated concept that was not mentioned before. To ensure that no irrelevant items were selected, we only used the five items with the highest activation. The probability of a concept to be selected is proportional to its activation. The concept is selected from one instance and activated in the other, where a new spreading process is initiated. By repeating this process in turn for both JAM instances, a simple dialog on a concept level is generated. In [14] we proposed a simple method how to select utterances from a set of templates which optimally transport the selected items. This would allow to transform the sequence of stimuli into a real conversation, given additional effort for generating the templates.

Examples of such dialogs on a concept level are shown in Tables 2 and 3. These conversations could have started with a discussion on fruits, leading to the stimulation of the concepts "apple" and "orange". The left column contains the concepts that were selected from the first instance and stimulated in the second, the right column shows the concepts that were selected from the second instance. Table 2 shows a conversation going back and forth without outside intervention. In table 3, we introduced new external stimuli, which were activated in both JAM instances, halfway through the dialog. The new stimuli "tropic" and "island" could for example be caused by passing a holiday advertisment during the conversation. The rest of the conversation is influenced by the newly introduced topic.

We can see that most of the concepts in the conversation are related to the originally stimulated items, i.e. we maintain a coherent conversation. Still, we see a gradual shift of topic. This is especially true for the second example, where the new stimuli strongly influence the course of the interaction. Note that the new items do not simply override the old ones. The spreading process supports items which are associated to both the old and the new concepts, e.g. tropical fruits. These examples indicate that JAM is indeed able to generate coherent interactions over a period of time, including the handling of gradual topic drift.

It has to be noted that the quality of the conversation depends on the quality and quantity of relevant nodes in the knowledge base. For certain domain specific parts of an application, it will be necessary to manually extend the existing entries.

Table 2 JAM conversation

Instance 1	**Instance 2**
Initial stimuli: apple, orange	
juice	
	lemon
make lemon butter	
	fruit
sour	
	store
bitter	
	sugar
lime	
	drink
lemonade	
	refrigerator
seed	
	citrus fruit
pineapple	
	yellow
bergamot	
	yellow fruit
pear	
	sweet

Table 3 Topic switch

Instance 1	**Instance 2**
Initial stimuli: apple, orange	
juice	
	tangerine
citrus fruit	
	lime
like orange	
	eat orange
peel orange	
	outside
sweet	
	bergamot
New stimuli: tropic, island	
ficus	
	pineapple
tropical fruit	
	person
peel first	
	banana
yellow	
	mango
salsa	
	fun

References

1. J.R. Anderson, D. Bothell, Michael D. Byrne, S. Douglass, C. Lebiere, and Y. Qin. An Integrated Theory of the Mind. *Psychological review*, 111(4):1036–60, October 2004.
2. J.R. Anderson and L.M. Reder. The fan effect: New results and new theories. *Journal of Experimental Psychology: General*, 128(2):186, 1999.
3. Thomas D. Erickson and Mark E. Mattson. From words to meaning: A semantic illusion. *Journal of Verbal Learning and Verbal Behavior*, 20(5):540–551, 1981.

4. C. Havasi, R. Speer, and Jason B. Alonso. ConceptNet 3: a Flexible, Multilingual Semantic Network for Common Sense Knowledge. *Recent Advances in Natural Language*, 2007.
5. B. Iordanov. HyperGraphDB: a Generalized Graph Database. *Web-Age Information Management*, pages 25–36, 2010.
6. A. Jameson, R. Schäfer, T. Weis, A. Berthold, and T. Weyrath. Making systems sensitive to the user's time and working memory constraints. In *Proceedings of the 4th international conference on intelligent user interfaces*, pages 79–86. ACM, 1998.
7. P. Kenny, A. Hartholt, J. Gratch, W. Swartout, D. Traum, S. Marsella, and D. Piepol. Building interactive virtual humans for training environments. In *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*, volume 2007. NTSA, 2007.
8. K.R. Koedinger and A. Corbett. Cognitive tutors: Technology bringing learning sciences to the classroom. *The Cambridge handbook of the learning sciences*, pages 61–78, 2006.
9. H. Lieberman, A. Faaborg, W. Daher, and J. Espinosa. How to wreck a nice beach you sing calm incense. In *Proceedings of the 10th international conference on Intelligent user interfaces*, pages 278–280. ACM, 2005.
10. Hu. Liu and P. Singh. ConceptNet - A practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226, October 2004.
11. C. Matuszek, J. Cabral, and M. Witbrock. An introduction to the syntax and content of Cyc. *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge*, pages 44–49, 2006.
12. T. Paek and E. Horvitz. Grounding criterion: Toward a formal theory of grounding. 2000.
13. O. Pietquin and T. Dutoit. A probabilistic framework for dialog simulation and optimal strategy learning. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(2):589–599, 2006.
14. F. Putze and T. Schultz. Utterance Selection for Speech Acts in a Cognitive Tourguide Scenario. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.
15. J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, 2007.
16. J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The Knowledge Engineering Review*, 21, 2006.
17. H. Schultheis, T. Barkowsky, and S. Bertel. LTMc An Improved Long-Term Memory for Cognitive Architectures. In *Proceedings of the Seventh International Conference on Cognitive Modeling*, pages 274–279, 2006.
18. M. Wolters, K. Georgila, J.D. Moore, R.H. Logie, S.E. MacPherson, and M. Watson. Reducing working memory load in spoken dialogue systems. *Interacting with Computers*, 21(4):276–287, 2009.