Knowledge Representation and Acquisition for Large-Scale Semantic Memory

Julian Szymański and Włodzisław Duch

Abstract—**Acquisition and representation of semantic concepts is a necessary requirement for the understanding of natural languages by cognitive systems. Word games provide an interesting opportunity for semantic knowledge acquisition that may be used to construct semantic memory. A task-dependent architecture of the knowledge base inspired by psycholinguistic theories of human cognition process is introduced. The core of the system is an algorithm for semantic search using a simplified vector representation of concepts. Based on this algorithm a 20 questions game has been implemented. This implementation provides an example of an application of the semantic memory, but also allows for testing the linguistic competence of the system. A web portal with Haptek-based talking head interface facilitates acquisition of a new knowledge while playing the game and engaging in dialogs with users.**

I. Introduction

Semantic memory is one of the key elements of the human cognitive system. It is a container for general knowledge about the world, storing associations between word meanings, providing systematic relations between concepts, working as a kind of mental lexicon deeply involved in all human language-related activities, and probably in all thinking and reasoning processes. The need to distinguish semantic memory from other types of memory has been argued for by Tulving [1] as a part of his theory of long term memory. Three psycholinguistic theories of the data organization in semantic memory have been proposed: a hierarchical model [2], spreading activation model [3], and prototype-based model [4]. They may serve as inspirations for the implementation of the linguistic competences in cognitive systems [5], [6]. Artificial intelligence based on symbolic approach failed to create decent natural language interfaces, while the subsymbolic approach [7], [8] has been successful only in psycholinguistic investigation, or in analysis of small-scale scripted stories [9], [10]. Natural language processing remains as one of the greates challanges to computational intelligence.

A system capable of understanding natural language should have a semantic lexicon that stores the meaning of basic concepts, and should be able to construct rich representations of the linguistic objects from this basic knowledge. A cognitive system with semantic memory containing concepts with well defined relations can be used to bootstrap itself, building and incrementally improving through natural language interactions with the system. With more linguistic information in the semantic memory, disambiguation of word meanings and classification of text episodes based on *a priori* knowledge is strongly improved [11], [12]. Construction of semantic memory requires two basic tasks to be completed: first, building its computational model based on flexible knowledge representation, and second, filling it with data.

In this article we shall further extend our previous attempts based on vector space representation of concepts [6] to solve these two issues. In the following section problems of knowledge representation for semantic memory is discussed. Several ambitious approaches to knowledge representation, such as the frame-based CyC expert system [13], use very complex knowledge structures, and therefore impede efficient evaluation of properties of the whole concept space. Much simpler vector representations of concepts are sufficient in most applications. In the third section the semantic search algorithm is introduced, and in the fourth section it is applied to the 20-question game. In the fifth section active learning aimed at improving data acquisition and verifying its correctness is presented. Experimental results are evaluated in section six, and the paper is closed with a discussion.

II. Knowledge representation for semantic memory

In word games and some other applications, the simplest representation of knowledge in form of concept description vectors (CDVs) is already quite useful [6]. CDVs are binary vectors with elements equal to 1 for features that can be used to describe an object, and 0 for all other features. Going beyond such simple representation "semantic networks", or networks of interconnected concepts, may be used [14]. In many types of such networks the basic elements are unchanged, taking the form of relations between triples: objects – attributes – values (OAV, [15]). There may be many different interpretations of such relations, for example RDF [1] forms triples: object – predicate – property. The set of interconnected words forms a semantic network of concepts, with connections identified with typed relations between them. Although ontologies are usually represented by hierarchical trees semantic networks may provide more information in some domains. They can easily be visualized in a graphical form. Using predefined relation types, semantic networks are able to display cognitive economy – features of concepts from the higher (more general) ontological levels can be inherited by more specific terms through the IS_A relation. In this way general knowledge for specific objects

Julian Szymański is with the Dept. of Electronic, Telecommunication and Informatics, Gdańsk University of Technology, Gdańsk, Poland (email: julian.szymanski@eti.pg.gda.pl), and Włodzisław Duch is with the Dept. of Informatics, Nicolaus Copernicus University, Grudziądzka 5, Toruń, Poland (contact – Google: W. Duch).

[1] see RDF primer at http://www.w3.org/TR/REC-rdf-syntax/

World Congress on Computational Intelligence (WCCI'08), Hong Kong, 1-6 June 2008, IEEE Press, pp. 3117-3124

978-1-4244-1821-3/08/$25.00©2008 IEEE

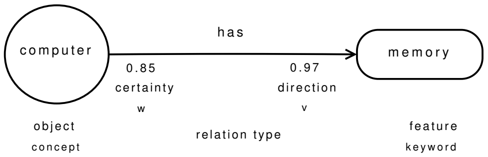

Fig. 1. Basic knowledge chunk vwCRK

may be obtained without the necessity of storing direct relations between each feature and each object.

However, such an approach has also some drawbacks: the inferences are computed through operations on the graph, limiting the flexibility of representations, for example preventing direct IS_A interpretation. To compute inheritance additional processing time connected with these type of relations is needed. There are also limitations for handling exceptions, ambiguities and problems with defining the (un)certainty of knowledge. To avoid these drawbacks triples that cannot express knowledge in form of object – has – an attribute – with a given value are extended here to knowledge chunks with information about certainty and strength of such relations. Sentences in the form of "object – is connected – using given relation – with the feature" may be modeled in this way. An example of atom of knowledge used in our system is shown in Fig. 1.

The basic knowledge chunk is here called vwCRK (*v w* weights, Concept – Relation – Keyword). This allows to form composite sentences of the form OAV or RDF. Factors *v* and *w* allow for flexible specification of a new (deduced) knowledge, with certainty estimated by probability factors, enabling also the handling of exceptions through implicit definitions for particular concepts and their features. The factor *w* estimates the strength of the relation (in most cases estimating how true or false it is), and the *v* factor its certainty. For example "the swan is white" gets the *w* value near +1, but the sentence "the swan is yellow" has *w* about -1. Both sentences have *v* value near 1, meaning that this knowledge is sure.

Knowledge in form of the vwCRK chunks is represented in a quite simple way enabling associative reasoning, but in some applications a large number of association should be evaluated. For example, to invent a good question that gives maximum information helping to understand the sentence, a lot of inferences will be needed. Therefore to facilited efficient evaluation of many concept relations the network of vwCRK elements is transformed to its equivalent vector representation. Conversion is based on the CDV concept description vectors – sets of features describing each of the objects that appear in the vwCRK semantic network. This results in a matrix with objects indexed by columns, and features indexed by rows. The vectors of CDV matrix obtained in this way will be very sparse, because most of the features are not directly connected to a given object. To discover all features that are applicable to a given concept procedural interpretation of selected relation types is introduced:

1) The **IS_A** relation allows to obtain specific features from more general objects; the inherited features obtain *w* values equal to the corresponding values of superior relations, and the *v* value is decreased by 10% and corrected during interaction with the human user.
2) The **Similar** relation type defines objects which share features with each other; this relation gives the opportunity to acquire new knowledge from similar objects through swapping of unknown features with given certainty factors.
3) The **Excludes** relation allows to exchange some unknown features (analogically to the "similar" relation), but the sign of the *w* weight value of the relation is reversed.
4) The **Entail** relation allows to acquire information about the applicability of additional features based on the information about connections between features; the presence of one feature automatically entails a few more features (connected via the entail relation) to appear in the CDV description of the object. This is analogical to the logical implication.

The smallest chunk (atom) of knowledge contains three components, defining the strength and the direction of relations between concepts and keywords: directly entered into the knowledge base (D), deduced using predefined relation types (V) from the stored information, and (A) obtained during system's interaction with the human user (see section V below). The weight w is determined from the average linear combination of these three components:

$$w = \text{avg}(\alpha D + \beta V + \gamma A) \quad (1)$$

where: α, β, γ are arbitrary chosen weights summing to one (for example, 0.25, 0.34, 0.41). The introduction of these parameters enables learning and correcting the knowledge base by obtaining new knowledge through interaction with users.

III. Semantic search algorithm

A lot of default knowledge is brought into language comprehension. Objects in the semantic network mapped into CDV space can be identified by referring to their features. In human minds these features invoke either specific objects or general categories of objects. Typical searching process requires matching a sufficient number of keywords to uniquely identify an object. To discover the minimum number of features needed for unique identification the system should interactively ask most relevant questions to gain maximum information. The answers received select the most probable sets of objects. This idea of selecting best features is the basis of knowledge representation by decision trees, but here we need to construct such trees in a dynamical way as the initial knowledge given to the system does not need to correspond to the top levels of any existing tree. Given the CDV matrix representation for a group of objects the fraction of all objects for which the keyword has value v_i is calculated as $p_i = p(keyword = v_i), i = 1..m$. Using this distribution of probabilities, Shannon information (IG) is defined for each feature for a given subspace of all objects:

$$IG(feature) = -\sum_{i}^{m} p_i \log p_i \tag{2}$$

The feature with the highest information value is used to formulate a question, and the answer to this question is used to reduce uncertainty about object identification. The initial description of the object and the subsequent user answers are collected in the answer vector A. This vector is used as a reference to calculate distance from the objects in the semantic space. In the n–dimensional subspace of all known feature values Euclidean distance can be used to evaluate which objects (represented by their CDV vectors) are most likely; for a given vector V:

$$d(V, A) = \sqrt{\sum_{i=1}^{n} (V_i - A_i)^2} \tag{3}$$

However, CDV vectors are very sparse because the detailed knowledge about objects is missing, and most features are not applicable to a given object, and the known features have different values of certainty (v) and strength (w). Therefore better results are obtained with the distance calculated according to the formula:

$$d(V, A) = \frac{\sum_{i=1}^{N} (1 - dist(V_i, A_i))}{N} \tag{4}$$

where N is the number of features, and the distance is calculated as

$$dist(x, y) = \begin{cases} 0 & \text{, if } y = NULL \\ -abs(y)/N & \text{, if } v = 0 \\ v|x - y| & \text{, if } v > 0 \end{cases} \tag{5}$$

where:
x - is the value of CDV component (weight w describing relation),
y - is the answer given by a human user for the successive questions. The NULL value denotes the “don’t know” answer obtained from the user.
v - is the certainty of knowledge in CDV.

The subspace of the most probable concepts lies in the minimum distance covering around the answer vector A:

$$O(A) = \{c | d(V(c), A) = \min\} \tag{6}$$

Here $d(V(c), A)$ is the distance between the object c represented by its CDV vector and the answer vector A.

Using an iterative process, the information in each feature is calculated with reduced number of objects from the $O(A)$ covering, and is used to formulate the next question. The fastest convergence to the solution is obtained by shrinking this covering, selecting at each iteration only the objects with minimal distance, but this approach does not take into account the possible mistakes (wrong, or inaccurate answers) of the user, and thus can cause the search to fail. This problem is discussed below.

Fig. 2. The Avatar used for the 20 questions game.

IV. The 20 questions game

The search algorithm presented above has been used to implement the 20 questions game. First, it shows an interesting application of semantic memory, and second, it allows for evaluation of the quality of collected knowledge. In this game the machine tries to guess the concept that a human player is thinking about. The questions are in the form that can be answered in a simple way: yes, no, seldom, sometimes and don’t know. The consecutive games should help to actualize the knowledge base and thus increase competence of the system. We are aware of only one computer implementation of this game[2]. In this implementation a set of about 500 fixed questions is used, and a table of objects/questions with weights associated with each answer has been built and is updated as a result of new games. Although the algorithm works quite well it does not contain real representation of objects, it is specialized only to play the 20 question game, while in our case the main goal is to create and test good semantic memory that has wide applications.

The user interface may be constructed in a form of an Avatar, a talking head simulated using the 3-dimensional graphics. The Avatar (Fig. 2) has been built using the Haptek technology [3]. It may use speech recognition and synthesis for communication, enabling natural communication; in our implementation standard Microsoft Windows tools (Speech API) have been used. It works quite well for the limited number of words that need to be recognized.

The game has been implemented and may be accessed at the web portal using the Avatar interface (but without speech recognition) at: http://diodor.eti.pg.gda.pl, or using a simple text interface at: http://diodor.eti.pg.gda.pl/simple.html

Implementation of the 20 questions game required a few modifications of the semantic search algorithm. First,

[2] see http://www.20q.net
[3] see the Haptek site http://www.haptek.com

the feature selected to formulate the question presented to the human user is based on its information value. Always choosing the best feature will lead to a situation where each time the search is repeated the same questions are being asked, as the algorithm goes on through the same search path. Although it is optimal in terms of convergence, it may be annoying and boring for the user. This is avoided by selecting keywords using a lottery, with a probability proportional to the information content of each feature:

$$p(c_i) = \frac{IG_i}{\sum_k^N IG_k} \tag{7}$$

Randomized questions allow to guess the same concept each time in a different way, making the play more interesting, but they can also lead to a larger number of questions. In the worst case the search algorithm could fail to find the right answer. To prevent this, one may restrict the selection to a few best keywords only, or alternate between maximization of information and randomized choice. Asking questions in such a randomized way makes not only the game more attractive, but allows also for more effective knowledge acquisition, as described below. Experiments show that alternating between optimal selection and the probabilistic choice of the next question is sufficient to find the object in the restricted semantic space used for testing in less than 20 steps.

The second modification of the semantic search algorithm concerns the metod of narrowing subspaces. Choosing only the objects that have a minimal distance to the answer vector at the next step of the algorithm does not take into account possible mistakes of the player, different points of view of different people, or some mistakes in knowledge base. To take this into account a soft "safety margin" is added to the minimal distance covering of the answer vector A. The probability p_{border} that an object is in the covering set of the answer vector in the next step of the game is estimated in a heuristic way assuming that $p_{border} = 1/d_{border}$, where:

$$d_{border} = d_{min} + \text{std}(O(A))/k \tag{8}$$

where d_{min} is the minimal distance between the answer vector A and its nearest CDV vector, and $\text{std}(O(A))$ is the standard deviation computed for the set of the objects in the $O(A)$ covering subspace, divided by k, the step number of the game, narrowing the margin as more information is collected.

The assumption that the game requires 20 questions is frequently excessive. In games with narrow domains concepts may be found even after a few questions, with additional questions asked only to confirm the certainty of the conclusions. In such a case it may be advantageous to make a quick guess of the concept. To determine when to make such a guess, a heuristic based on analysis of the distance from the most probable object and object next to it is used. If this distance grows with each new question asked, the algorithm makes a guess.

V. Active learning

Common sense knowledge can be represented in the form of relations between objects and features, stored in semantic memory. Three methods have been used to collect this knowledge:

1) manual editing,
2) importing information from ontologies and machine readable dictionaries,
3) data mining algorithms used on free text to search for statistical correlations between concepts.

These three approaches have been used to construct initial semantic memory, and then purify and extend it in an interactive dialogs with users. The semantic search algorithm is used here also for verification of the quality of the semantic memory: if the search process is finished with success, knowledge collected in the semantic memory is correct and useful. The 20 questions game may be used for verification and acquisition of additional knowledge for the semantic memory.

A. Learning new objects

The 20-question game system may guess the concept either correctly or wrongly. If the guess was correct the system knows enough about the object, the questions that have been asked and the answers obtained from the user were sufficiently informative. Results may be used to modify the CDV of the concept that has been guessed correctly updating features corresponding to questions with definite answers (that is, yes or no). The second scenario, when system fails to guess the concept correctly, is followed by asking an additional question: *What did you have in mind?* The parsed answer gives the right concept (in terms of the questions asked), and allows for addition of a new concept, or for a modification of the CDV vector representation of an existing concept. The modification of the knowledge is made according to the Eq. (1), where the A component is modified.

B. Learning new features

During the search process in the feature space some objects described by their CDV vectors may appear identical. In this case they cannot be correctly distinguished by asking questions, and the semantic memory system should acquire new knowledge to disambiguate identical concepts. In this situation, the scenario of a dialog for obtaining new features for non-separable concepts is invoked. For each of the n identical concepts the question is asked: *Tell me what characterizes the <concept>*. Parsed user answers allow to enrich CDV representation for the specified object, and allow the entry of new features into the semantic memory. They can then appear as one of the questions in the 20q quiz, and thus can enhance also representations of other concepts.

C. Extensions

Implementation of the new dialog scenarios should make the word game playing system more attractive, and should

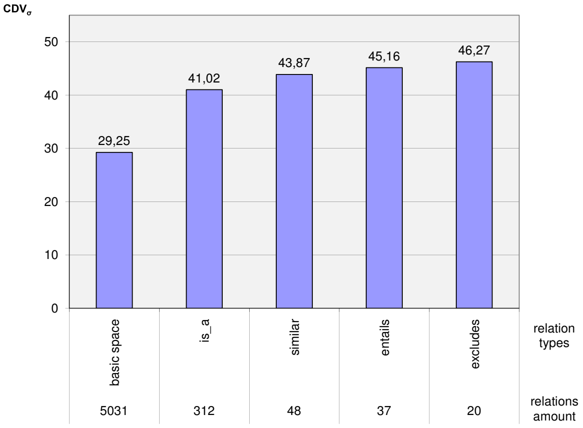

Fig. 3. CDV density changes according to relation type

allow for gathering knowledge in a more effective way. One of the new dialogue scenarios is aimed at generalization of properties, to keep the features at the most general and most useful level. In the process of collecting knowledge and converting it to the CDV representation a lot of redundant information is created: some features are defined multiple times at more specialized taxonomy levels. One can easily identify situations where one of the features is connected with objects of the subnode in the taxonomy tree. Identification of that feature allows to ask the player a question: *Is it true that <feature> is typical for <upper concept>?*

Positive answer allows to add features to more general concepts and propagate them to all more specific subnodes. This type of active dialog allows for restructuring of the semantic memory. More dialog scenarios may be used to obtain new specific knowledge, and correct or extend the knowledge already stored. It should be worthwhile to create dialogs for obtaining new knowledge using analogies between set of concepts, but this has not been done so far.

VI. Experimental results

To test the proposed approach in a restricted domain, semantic memory for the animal kingdom domain has been created (another application to diagnosis of mental disorders has also been created but is too specialized to be discussed here). The basic semantic space has been constructed as an aggregation of the knowledge from different machine readable knowledge sources: WordNet [16], ConceptNet [17], SumoMilo ontology [18], and the MindNet [19] project. To avoid irrelevant concepts and features the knowledge is added to the semantic memory only if it appears in two or more of these sources. The basic semantic space size contained 172 objects and 475 features involved in 5031 relations. Changing knowledge representation from a semantic network (vwCRK) into a simpler representation based on CDV vectors helps to obtain new features. Figure 3 presents the number of features per concept, called CDV density, after adding interpretation of different types of relations.

The first bar represents "pure" semantic space, with an average of 29 features per concept, features that have been obtained directly from the primary sources. The subsequent bars show how the number of features in CDV grows after applying particular types of relations: "is-a, similar, entails, excludes". After all these relations have been applied, an average of over 46 features per concept have been obtained.

The semantic search algorithm can be used to measure the quality of the semantic memory. The coefficient Q that estimates this quality may be taken as the proportion between the number of searches that finished with success N_S and the total number of searches N. The semantic memory error is $E = 1 - Q = 1 - N_S/N$.

The quality of the basic semantic space has been calculated for 10 random concepts. To obtain more reliable estimate, probability distribution of the number of features in CDV has been used to choose concepts with the number of features close to an average. For the animal domain selecting 10 random concepts gives Q around 0.8, and repeated 5 times this process gave the average error $E = 0.18$.

It is also instructive to evaluate the speed of learning of new concepts. This may be estimated as the average number of games and search steps that have to be made to correctly recognize new concept. Because each finished game changes the state of the knowledge base, search of a new concept is made interchangeably with its two most similar concepts, demonstrating the increase of separability of the new concept. Search of all new concepts and two additional (the most similar) makes one step of test procedure. Initially the test procedure has been run for 30 concepts (the number of additional runs was taken as 3 times the number of failed concepts, making it depended on the result of the search process). All 10 new concepts have been learnt after 5 steps of test procedure, but it should be mentioned here the last two steps were performed only for one concept search.

For the initial semantic space the average number of games played until the system correctly recognized the new concept was $N_f = 2.79$. The precise value for a given concept depends on the number of semantic neighbors that are close to this concept. More games have to be played if the new concept is quite similar to many other concepts, while for distinct new concepts sometimes a single game will be sufficient. The average number of search steps (questions) is proportional to N_f.

Another factor that is useful for estimation of the semantic memory quality is the completeness of concept representation. This has been evaluated in two ways. First, the CDV description of the concept is considered sufficiently detailed if the concept can be guessed playing the game. Second, the number of features defined for the concept may be compared to a golden standard, that is a manually created predefined set of features that are relevant for the concept. Four measures of the concept description quality have been introduced for each concept O.

1) $S_d = N_f(GS) - N_f(O)$ is the measure of incompleteness. $N_f(GS)$ is the number of features defined

in the Golden Standard $GS = G(O)$ for the concept O, and $N_f(O)$ is the number of features defined for this concept in $CDV(O)$. The S_d value shows how many features are still missing compared to the golden standard.

2) $S_{GS} = \sum_{i=1}^{N_f(O)} [1 - \delta(CDV_i(GS) - CDV_i(O))]$ is the measure of similarity, based on the co-occurrence of features; the sum is only over features with defined yes/no values. The S_{GS} value is the number of features from O that are found in the golden standard GS vectors; the reverse measure (S_{NO}) is defined below. The ratio S_d/S_{GS} of the similarity and incompleteness measure shows the percentage of all features of the golden standard that has already been defined for the concept O.
3) $Dif_w = \sum_{i=1}^{m} (|CDV_i(O) - CDV_i(GS)|/m$ is the average difference for all m feature values that appear in both O and GS representations. This measure shows how the feature values differ in O and GS vectors for those features that are common to the two vectors.
4) $S_{NO} = \sum_{i=1}^{N_f(GS)} [1 - \delta(CDV_i(O) - CDV_i(GS))]$, analogically to S_{GS}, is the measure of similarity of two CDV vectors based on co-occurring features, with summation running over features in the GS. The S_{NO} value is equal to the number of features that appear in description of the concept O and are not found in the GS.

The difference between $CDV(O)$ and GS representations is not only due to the lack of knowledge, but may also come from implementation of the mechanism to partially randomize questions, allowing for more knowledge acquisition when the game with the same concept is repeated several time.

The quality measures defined above have been evaluated in the following way:

1) A concept O is chosen randomly with the probability proportional to the $\exp(N(O)/N))$, where $N(O)$ is the number of features in concept O and N is the total number of features, giving concepts with larger set of features a higher chance of being selected.
2) The $CDV(O)$ representation of the chosen concept O is inspected, and if necessary corrected.
3) The $CDV(O)$ is removed from the memory.
4) The system tries to learn the concept O by playing the 20 questions game.

Average results of performing this procedure for 5 test objects are depicted in figure 4, illustrating the changes of different quality measures as the learned representations are slowly approaching the desired golden standards. The dynamics of the process is shown as a function of the number of games played.

The results show that the active learning method is useful for gaining new features for concept description vectors – the CDV representation of new concepts is getting more precise, increasing the number of well-defined features. This is clear from the NO_ρ ($NO_\rho = S_{NO} + S_{GS}$) graph showing the average growth of the number of features as a function of the number of games played. Randomization of questions helps to find different features in each game. For the tested set of concepts, the average number of the games was $N_f = 2.67$. After the first successful game when a particular concept has been correctly recognized it was always found properly.

The value S_d is calculated for the average number of features of the golden standard of all 5 tested objects. For the five concepts used in test this was 55.5 features. The decreasing trend implies that the features from the new CDV vector progressively cover the features of the golden standard. It may achieve zero or even become negative if more games are played (thanks to the randomization of questions), but the speed of convergence is asymptotically decreasing to zero, because with each new game the number of new features acquired is decreasing.

The S_{NO} values represent the number of features that have not been covered by the golden standard. The number of such features is near 30% of all features in the new CDV vector. This has no influence on the quality of searching as reflected in the stabilization of the $N_f = 2.67$ value. It simply shows that a significant part of the 475 features that define the whole semantic space are irrelevant to the golden standard representation of a particular concept even in such restricted domain.

The covergence of coverage to the golden standard is shown in the graph of S_{GS}. After 4 games only a few new fatures are added. Decreasing speed of feature acquisition shows that in the later games more features will be out of this set, so the percentage of S_{NO} in NO_ρ will grow.

VII. Discussion and future plans

Artificial Intelligence has failed to deliver its promises in many areas, but arguably the most disappointing failure has been in the language domain. There is a strong tendency towards embodied cognitive systems, hoping that somehow higher cognitive functions will emerge with internal representations related to the action-perception information processing. This trend is important and interesting, but it remains to be seen whether symbolic understanding and reasoning may be achieved in this way. Aaron Sloman (University of Birmingham, private information) has recently argued that cognition is tethered rather than embodied, and therefore one should not abandon simpler approaches to natural language processing. A lot has yet to be done, for example there are no common-sense ontologies, or semantic networks that will systematically describe simple concepts, or even simplest vector representations of most common concepts.

The approach presented here goes in the direction of combining semantic network representation with simplified CDV vector space representation to facilitate fast searches and retrieval of information. This is not sufficient for full text understanding required in a dialog system, but is a big step forward compared to the purely symbolic template matching techniques [20] used by chatterbots. The use of CDV representation enables concept definition, evaluation of concept similarity and refinement of queries in ambiguous situations, significantly increasing the level of competence of

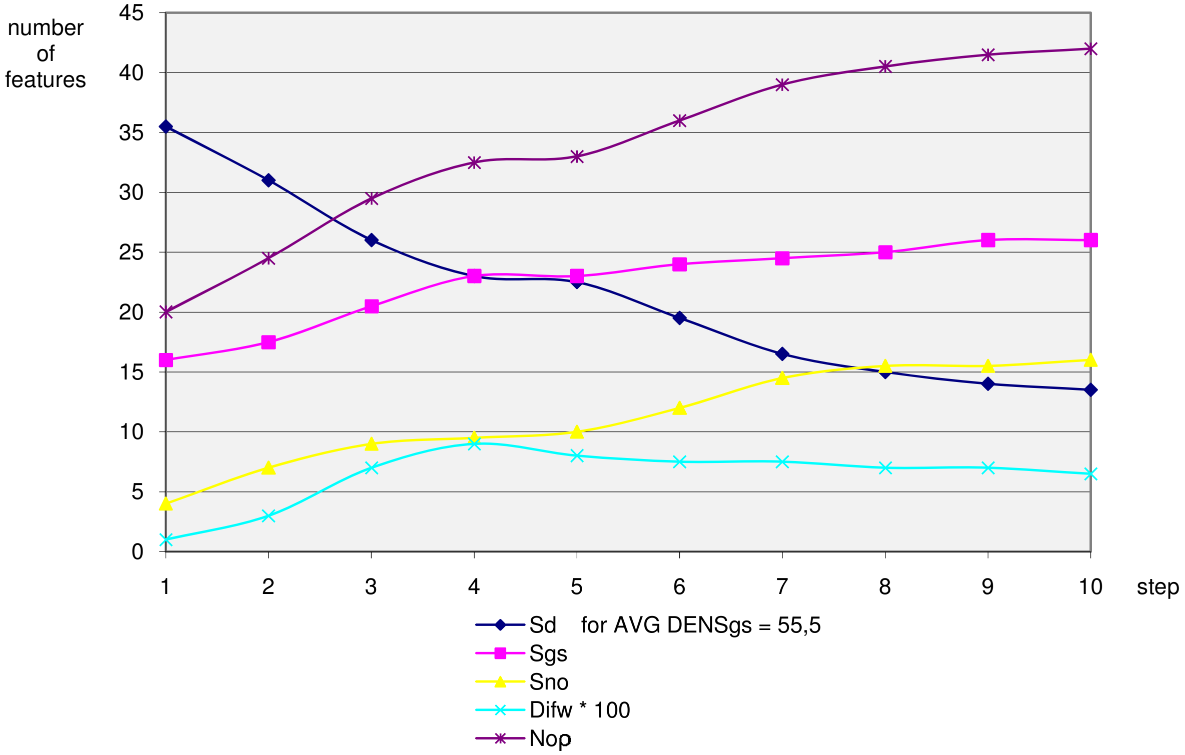

Fig. 4. Average changes of concept description vectors

linguistic systems. In particular, various word games may be played using semantic memory with avatar-based interfaces in a natural way. Winning the 20 question game against humans in an unrestricted domain is an important challenge for computational intelligence, perhaps more significant than winning with the chess world champions. In chess speed of computing in very important, but in word games extensive knowledge about language concepts is needed. Other applications of semantic memory include search engines that should be able to query the user for precise meaning, or chatterbot systems that ask intelligent questions using limited representation of concept properties.

The biggest problem is how to collect this knowledge in an automatic or semi-automatic way. Although we have used many sources (WordNet [16], ConceptNet [17], SumoMilo ontology [18], MindNet [19]), and have restricted the domain to animals and their features, the knowledge that may be generated in this way [6] is still far from what an average human knows about the subject. Additional knowledge may be gained from such sources as the Wikipedia articles or CyC ontologies. These sources of knowledge may create initial semantic memory, and the active learning and dialog scenarios discussed here may help to collect more knowledge.

It is not enough to determine similarity between the concepts by comparing their CDV representation, as small differences may sometimes be sufficient to distinguish between concepts and different subsets of features may be used to discriminate concepts. The guessing game with randomized questions played between two programs could be the key to identify concepts that are hard to guess and thus their representation should be changed. An active search for more knowledge to discriminate between similar concepts, combined with dialogues with the curators of the semantic memory should also help to add and correct the new knowledge. In the end a large-scale collaborative effort, similar to Wordnet or MindNet, may be needed to remove remaining errors and provide useful enhancements. We are working on such system now, and hope that it will allow for a significant progress in natural language processing capabilities of cognitive architectures.

Acknowledgment

W.D. is grateful for the support by the Polish Committee for Scientific Research, research grant 2005-2007 and J.S. has been supported in part by the grant no. N516 035 31/3499.

References

[1] E. Tulving, “Episodic and semantic memory,” *Organization of memory*, pp. 381–403, 1972.

[2] A. Collins and M. Quillian, “Retrieval time from semantic memory,” *Journal of Verbal Learning and Verbal Behaviour*, vol. 8, pp. 240–247, 1969.

[3] A. Collins and E. Loftus, “A spreading-activation theory of semantic processing,” *Psychological Review*, vol. 82, no. 6, pp. 407–428, 1975.

[4] E. Rosch, “Categorization of natural objects,” *Annual Review of Psychology*, vol. 32, pp. 89–115, 1981.

[5] L. Itert, W. Duch, and J. Pestian, “Medical document categorization using a priori knowledge,” *Lecture Notes in Computer Science*, vol. 3696, pp. 641–646, 2005.

[6] J. Szymanski, T. Sarnatowicz, and W. Duch, “Towards avatars with artificial minds: Role of semantic memory,” *Journal of Ubiquitous Computing and Intelligence*, vol. 1, in print.

[7] D. Rumelhart and J. M. (eds), *Parallel Distributed Processing, Vol. 1: Foundations*. Cambridge, MA: MIT Press, 1986.

[8] S. Lamb, *Pathways of the Brain: The Neurocognitive Basis of Language*. Amsterdam & Philadelphia: J. Benjamins Publishing Co, 1999.

[9] R. Miikkulainen, *Subsymbolic Natural Language Processing: An Integrated Model of Scripts, Lexicon, and Memory*. Cambridge, MA: MIT Press, 1993.

[10] ——, “Text and discourse understanding: The DISCERN system,” in *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*, R. Dale, H. Moisl, and H. Somers, Eds. New York: Marcel Dekker, 1999.

[11] P. Majewski and J. Szymański, “Text Categorization with Semantic Commonsense Knowledge: First Results ,” ICONIP 2007, in print.

[12] W. Duch, P. Matykiewicz, and J. Pestian, “Towards understanding of natural language: Neurocognitive inspirations,” *Lecture Notes in Computer Science*, vol. 4669, p. 953–962, 2007.

[13] D. Lenat, “Cyc: A large-scale investment in knowledge infrastructure,” *Comm. of the ACM*, vol. 38, pp. 33–38, 1995.

[14] J. Sowa, Ed., *Principles of Semantic Networks: Explorations in the Representation of Knowledge*. Morgan Kaufmann, San Mateo, CA, 1991.

[15] G. Curtis, *Business Information Systems*. Addison-Wesley Inc, Boston, MA, USA, 1995.

[16] G. Miller, R. Beckitch, C. Fellbaum, D. Gross, and K. Miller, *Introduction to WordNet: An On-line Lexical Database*. Cognitive Science Laboratory, Princeton University Press, 1993.

[17] H. Liu and P. Singh, “Conceptnet. a practical commonsense reasoning tool-kit,” *BT Technology Journal*, vol. 22, no. 4, pp. 211–226, 2004.

[18] K. Ahrens, S. Chung, and C. Huang, “From lexical semantics to conceptual metaphors: Mapping principle verification with wordnet and sumo,” 2004, pp. 99–106.

[19] S. Richardson, W. Dolan, and L. Vanderwende, “Mindnet: acquiring and structuring semantic information from text,” 1998, pp. 1098–1102.

[20] R. Wallace, *The Elements of AIML Style*. ALICE A.I. Foundation, see www.alicebot.org, 2003.