Semantic Memory Knowledge Acquisition Through Active Dialogues

Julian Szymański, Włodzislaw Duch

***Abstract*— Many language-oriented problems cannot be solved without semantic memory containing descriptions of concepts at different level of details. Automatic creation of semantic memories is a great challenge even for the simplest knowledge representation methods based on relations between concepts and keywords. Semantic memory based on such simple knowledge representation facilitates implementation of quite interesting linguistic competences that have not yet been demonstrated by more sophisticated rule or frame-based knowledge bases, for example CYC. These linguistic abilities include word games, such as the twenty questions game, that may be implemented using semantic memory built on relational model for knowledge representation. Creation of large-scale knowledge base for semantic memory involves mining structured information sources (ontologies, dictionaries, encyclopedic entries) and free texts (textbooks and internet sources). Quality of this knowledge may be improved using collaborative projects in which systems that already possess some linguistic competence actively interact with human users, mining their knowledge. In this article three dialog scenarios for mining human knowledge are introduced, and the data acquired into semantic memory structures through such interaction is described.**

I. Introduction

Semantic memory is one of the key elements of the human cognition processes. It adds meaning to the concepts in memory, encoding their features, mutual relations, and facilitating associations between them. There are two psycholinguistic models of semantic memory. First, Collins and Quillian [1] introduced a hierarchical model that organizes concepts in taxonomies using sub-class and super-class relation types. Additionally each of the concepts has its own related features. The meaning of the object is formed by describing its features, including features from the upper taxonomy relations. In the second model, introduced by Collins and Loftus [2], concepts form conceptual relations network. The meaning of the concept is formed through the spread of activation from the concept that is being analyzed (and is thus fully active) to the concepts closely related to it. Although this is a dynamical process, capturing the activity of connected concepts after a few steps of spreading activation gives high-dimensional vector space representation of the concept. If the network contains all relevant connections this representation will carry more information than representations derived from statistical analysis of context windows.

Our computational model of semantic memory contains elements of both hierarchical and spreading activation approaches [3]. For some applications we have found that efficient knowledge representation may be based on the simplest concept description vectors (CDVs) that contain binary information indicating which properties are relevant to the description of the concept. In other applications a tertiary description: true, false or irrelevant, may be needed. In this paper the weighted Concept – Relation – Keyword (wCRK) representation is taken as the atomic unit of information to be stored in the semantic memory data structures. The set of wCRK forms knowledge stored in the semantic memory.

Concepts and keywords linked by relations create semantic network [4] that captures the knowledge base of the system. Although representation of knowledge by semantic networks has some limitations in comparison to the frame-based representations (e.g. CYC [5]), it is quite sufficient for many applications. The main problem is the lack of knowledge: there are no knowledge bases that will list all keywords relevant to a given concept. The situation is a bit better in biomedical applications, where thanks to the efforts of the US National Library of Medicine the Unified Medical Language System (UMLS) was created, containing information from 88 medical knowledge sources, defining over one million concepts relevant to medicine [6]. This information may be used to create spreading activation networks for disambiguation of medical concepts [7]. It is safe to say that many problems in natural language processing cannot be solved without an access to a large knowledge base for common sense reasoning. The complexity of CYC frame-based concept description prevents it from serving as associative memory, for example generating good questions in word games.

Knowledge for semantic memory may be derived from many sources: the WordNet lexicon [8], ConceptNet [9] – a commonsense knowledgebase generated from the large corpus of about 700,000 sentences collected in the collaborative project (with over 200,000 useful assertions), ontologies such as the Suggested Upper Merged Ontology (SUMO) and its domain ontologies (about 20,000 terms and 60,000 axioms) [10], and active free text search for features that may be applied to a given concept [3]. Despite all these efforts the quality of semantic network constructed in an automated way is still rather poor: it does not contain many relations that are obvious to humans – such relations are rarely found in the texts, forming implicit knowledge – and it may contain

Julian Szymański is with the Department of Electronic, Telecommunication and Informatics, Gdańsk University of Technology, Gdańsk, Poland; email: julian.szymanski@eti.pg.gda.pl

Włodzislaw Duch is with the Department of Informatics, Nicolaus Copernicus University, Toruń, Poland, and was in School of Computer Engineering, Nanyang Technological University, Singapore when writing this paper; Google: Duch

many relations that are false.

This paper is focused on the collaborative methods of knowledge mining aimed at the improvement of the semantic memory. We assume that the main part of the text mining has already been finished [3] so that sufficient knowledge has been acquired to function in an imperfect way. Further improvements will be made by actively involving the system – an avatar coupled to semantic memory that is asking questions and recording the answers – in a dialog with people playing some word games. In the next section one algorithm for such applications, the 20 questions game, is briefly described. Three scenarios used in an active dialogue are presented in section three. Section four contains evaluation and discussion of the knowledge gained in this way. The initial experiments described here have been narrowed down to a single domain of concepts related to the animal kingdom. This facilitates concentration on a single topic during the whole dialog. However, the approach is quite general and can be used in a large-scale collaborative project in any domain.

II. 20 Questions Game

Knowledge representation as named links between keywords and concepts facilitates various linguistic applications, including word games. The ability to capture semantic context of the utterance in word games requires a restricted linguistic competence that may be implemented in semantic memory. As an exaple we chosen an algorithm to play twenty questions game because it may serve as a paradigm for all problems requiring progressive approximation of the meaning of the query.

In the twenty question game computer program tries to guess a concept the opponent is thinking about, asking questions that should be answered only with "yes" or "no", although in some versions more answers are acceptable, including "irrelevant", "partially" or "sometimes". Each time a question is formulated a large space of concepts has to be analyzed. To increase the efficiency of this process semantic network should be replaced by simple numerical representation of relevant information defining the concepts that may be used in the game. To achieve this concepts will be represented in vector spaces using concept description vectors (CDV), a numerical representation of concepts using keywords (features) that characterize each concept. The set of the CDV vectors forms semantic space matrix. Using numerical representation for semantic space it is easy to perform numerical calculations on it. A distinction is made between concepts that one can use in the game, and properties of the concepts that may serve as keywords but cannot be the object of the game.

The twenty questions game algorithm should find the most informative feature and use it in the first question. It is done by calculating the Shannon information for each keyword according to:

$$I(keyword) = -\sum_{i}^{k} p_i \log p_i \tag{1}$$

$$p_i = p(keyword = v_i) \tag{2}$$

where $p(keyword = v_i)$ is a fraction of concept vectors for which the keyword has value v_i. The keyword that maximizes this measure should divide the space roughly in two equal subspaces. If there are several keywords that are nearly equivalent (have similar entropy) one should choose the most common word; information about frequency of word usage is available [12], [13]. This will increase the chance of correct response. Some keywords allow for completely unambiguous answers, but in general it is rather difficult to estimate.

The answer of a human player to the question based on the selected keyword allows for creation of a subspace $O(A)$ containing the set of the most probable concepts:

$$O(A) = \{k; \min ||CDV_k - A||\} \tag{3}$$

where A is a partial vector of retrieved answers, and CDV_k is a vector representing concept k. The iterative process for calculating information gain that selects from the whole space successive subspaces that shrink with the growing vector of retrieved answers leads to a guess in the twenty questions game, or to a unique set of no more than 20 questions that uniquely define each concept. The game gives an opportunity for estimation of the quality of knowledge stored in semantic memory. It is an interesting test for machine intelligence, as any program that has the ability to ask relevant questions and guess what people have in mind at the human level of competence should be called "intelligent". The linguistic competence of the program critically depends on the quality of its semantic memory, therefore playing successful games allows for verification of already possessed knowledge and the games lost, are an opportunity to learn new or correct existing relations. Mining human knowledge in this situation requires an active dialog. This is described in details in the next section.

III. Knowledge Acquisition Using Active Dialog

Basic knowledge for the semantic memory may be imported from external resources such as machine readable ontologies and dictionaries. In the experiments described below WordNet [8], ConceptNet [9] and SumoOntology [10] have been used. The concatenation of data from these sources created the initial semantic space. In the semantic space restricted to animal kingdom domain 94 concepts, 72 keywords and 6 types of relations have been used, leading to a semantic space with a total of 359 relations between concepts and keywords.

The data acquired using automatic text analysis may be inconsistent and therefore should be verified and extended. This can be done using manual graphical editor to browse through the semantic space, correcting and adding new relations and keywords. It is a tedious task and what is more important it is almost impossible to recall all relevant information. It is much easier to correct or remove superfluous information that may be found in the semantic memory. This does not solve the problem of missing data that should be added by the user. The extension and verification of this data will be done involving the user in a dialogue. This dialog is based on templates prepared for solving different problems with semantic memory – the lack of, inconsistence, or the need for corrections of the semantic memory knowledge. Solving these problems extends and improves knowledge base of the system. When no such issues arise any longer and the system plays well within its limited domain of competence it may be considered as intelligent. Identifying problems, and actively searching for new knowledge until a class of problems is correctly solved, requires intelligence that is more than just the ability to use stored knowledge [11].

The active dialog for clarifying and acquiring new knowledge is built using predefined sentence templates and existing semantic knowledge stored in memory. The idea is to ask human user questions that should help in acquiring and verifying knowledge. Dialogs are realized according to the following schema:

Phase I – formulation of the question.

- Identification of the semantic memory element. At this stage system identifies specific issues with semantic memory and relevant objects (concepts, keywords) of interest.
- Sentence generation using predefined templates. The chosen template depends on the method used for semantic memory element identification.
- User sentence presentation – asking the user.
- Processing (parsing) user answers – building new knowledge in the form of weighted concept-relations-keywords (wCRK). To prevent complications with parsing user answers should be given in simple sentences.

Phase II – verification and storage.

- Sentence generation using new wCRK.
- Asking for verification of newly acquired knowledge.
- Storage in semantic memory structures after successful verification.

Different methods for identification of semantic memory elements give the ability for building rich sets of dialog scenarios. Each of the methods of the identification represents the way for solving a specific issue. Below three dialog scenarios for solving some issues related to data stored in the semantic memory structures are presented.

A. Acquiring new keywords for a specified concept

The first scenario presented here is a basic template used when the system needs more knowledge for some concept. Several methods for identification of such concepts may be used, depending on the particular problem. For example, in some subspaces two or more concepts may become indistinguishable because insufficient number of keywords is available to characterize them. Should more keywords be added to concepts that are high in the ontology and are usually separated using first few questions, or should keywords be added to specific concepts that are left in subspaces near the end of the game? On the one hand it is beneficial to have good descriptions for more general concepts because these features are inherited through the is_a relation by more specific concepts, generating thus more knowledge. On the other hand characterization of specific concepts is also needed in terms of the twenty questions game. The system needs to know how to distinguish among different type of wildcats, and this requires very specific knowledge. Obviously there is no need for excessive number of keywords if the data could be correctly separated with smaller number of judiciously selected keywords.

The strategy for selecting interesting concepts is to create for each concept a measure of semantic information content (Si = semantic information). It can be calculated as:

$$Si = Cr * \frac{1}{1 + \sum \frac{1}{Kr}} \tag{4}$$

where Cr is concept popularity and Kr is popularity of keywords that describe it, equal to the normalized number of concepts the keyword relates to. Cr is computed as follows:

$$Cr = \frac{BNC * Grank * IC}{max(Cr)} \tag{5}$$

where *BNC* is a measure of word popularity taken from the British National Corpus [13], *Grank* is normalized number of pages containing the word returned by the search engine Google, and *IC* is the information content equal to the number of word instances in the WordNet descriptions. The Cr measure allows for choosing the candidate concepts for the dialog. In case of equal rank more general concept (higher in ontology, linked through is_a relations) should be chosen.

The sentence templates for this dialog scenario are:

I know that <concept> <relation type, keywords>[]. Can you tell me more about the <concept>?

In this dialog template <relation type, keywords>[] denotes a list of features related to the concept, using specified relation type. The conjunction 'and' stands as a separator for the list of concepts, while <concept> denotes concept of interest. This template is used when the concept has associated keywords.

The second template is:

I don't have any particular knowledge about <concept>. Can you tell me more about <concept>?

This template is used when the concept has no keywords.

Example

The selected concept using Cr measure is elk.

Computer: *I don't have any particular knowledge about elk. Can you tell me more about elk?*

Human: *Elk is a mammal.*

After parsing the answer system extracts knowledge wCRK(elk - is_a - mammal). This information has been already stored in semantic memory, thus system reports current state of knowledge and asks for additional information. The knowledge report is realized using the dialog template:

I know that <concept> <relation type> <keyword>. <concept> <relation type, keywords>[]. Please provide more knowledge about elk.

The knowledge displayed is limited to a particular concept relations, however checking is preformed on the whole CDV.
Computer: *I know that elk is mammal and elk is herbivore. Please provide more knowledge about elk.*
Human: Elks have antlers

This paring gives new knowledge wCRK(elk - has - antler). Verification step is performed next and after successful check new data is stored in the semantic memory.

B. Separability of Concepts

Concepts stored in the semantic memory can be represented in the CDV matrix. Using this method for data representation it is easy to determinate non-separable (identical) concepts in the semantic space. The next semantic memory check is to ensure that all concepts differ. The sentence template for this dialog scenario is:
<concept1> and <concept2> seem identical. Please give me the feature that may distinguish them.

Obtaining answer to such question will either introduce a new feature or change one of the value of existing feature, allowing for separation of semantic memory objects with identical CDVs. The aim of this dialog is similar as in previous section, but the dialog templates differ because here the new knowledge involves two concepts.

Example

The concepts panther and lion has identical CDV representation. The dialog below shows how the system gathers new knowledge for disambiguating them:
Computer: *panther and lion seems be identical. Please give me the feature that may distinguish them.*
Human: Lion has a mane.
Parsing user answer gives new knowledge wCRK(lion - has - mane). It is verified using query:
Computer: *Is it correct knowledge: lion has mane?*

The query confirmation built from the knowledge acquired during dialog causes semantic memory actualization. It is implicitly assumed that the second concept does not have the new feature. At this point it is useful to ask: what other objects that are in the same branch of the taxonomy tree also have the same feature?

This dialog scenario gives the opportunity to improve separability of concepts in the semantic memory. The advantage of completing semantic memory in this way is particularly clear during the final stage of the twenty questions game. The existence of non-separable concepts prevents the program from guessing precise object while thanks to new keywords questions that narrow the semantic sub-space to a single concept are generated. In case of sentence parsing failure system reports parsing problems, and asks human user for simpler and more specific answers.

C. Learning new concepts

The aim of the dialog scenario presented below is to obtain new concepts. New concepts come from the dialog with the user after the system fails to guess the concept user has been thinking of. In that way the system easily obtains knowledge about most popular concepts it did not have initially. When the game fails to guess the concept although it is stored in the semantic memory the knowledge has to be verified. Verification is performed for keywords related to that concept.

Example

The system asks for a concept it did not manage to guess.
Computer: *I give up. Please tell me what did you think of?*
Human: *It was duckbill.*
The program didn't find this concept in its semantic memory, it is trying to add new knowledge using a dialog template:
A <concept> is new to me. What is it?
The new concept should be fit in the right place in the taxonomy tree. It assumes that the parsing result should be the wCRK containing is_a relation type.
Computer: *A duckbill is new to me. What is it?*
Human: *It's an animal.*

The parsing stage formats this knowledge as wCRK(duckbill-is_a-animal). The animal has subclasses in taxonomy tree and they are not leaves. So this entry should be verified using sentence template:
Is <concept 1> a typical <concept 2> or tell me the name of <concept 2> sub-class <concept 1> belongs to.
Computer: Is duckbill a typical animal or tell me the name of animal sub-class the duckbill belongs to.
Human: It is an egg-laying mammal.

System obtains new knowledge wCRK(duckbill - is_a - egg-laying mammal) and wCRK(duckbill - is_a - animal). After verification the *egg-laying mammal* is stored as sub-class of *animal* and the *duckbill* is stored as sub-class of *egg-laying mammal*. After fitting new concept into the ontology the program is trying to obtain more details about this concept. It is done using scenario presented in the first section of the active dialog templates.

IV. Results

A sample game below is present to illustrate how active learning algorithm modifies CDVs. It demonstrates how the system completes very limited knowledge about concepts obtained from machine-readable structured sources. The user answers are marked by the [] brackets. The strength of the program in playing the twenty question game depends strongly on the quality of concept descriptions stored in CDVs, therefore the improvement in the quality of knowledge representations is evaluated after active dialogs. Let us assume that the player thinks about a 'fox'. Original CDV for 'fox' includes: {(is_a) mammal, (is_a) carnivore, (is_a) canine, (has) paws}. The first attempt at guessing this concept was: Is it a mammal? [YES]. Is it a carnivore? [YES]. Is it a canine? [YES]. Is it a dog? [NO]. At this point the program fails to guess the concept because there are two identical objects in the semantic space, 'hyena' and 'fox'. In the final part of the game objects may be guessed directly, but here the point is to use such situations as an opportunity to gain useful knowledge. The game adds new knowledge wCRK(hyena,is not, dog) and wCRK(fox, is not, dog), but separability scenario should still be run: "Hyena and fox seems to be identical”; “Tell me what is characteristic for hyena?” [it is scavenger], “Tell me what is characteristic for fox?” [it has bushy tail]. This extension of knowledge will let the program correctly guess the concept ‘fox’ adding new query to the game: Is it scavenger? [NO]. I guess it is a fox.

The above example show how games help to extend knowledge stored in the semantic memory. The number of failed games may be used to estimate how useful the knowledge added from the active dialogs is in comparison to the initial knowledge stored in semantic memory after analysis of structured machine-readable source. Selecting 15 new concepts on average about 3.9 failed games followed by active dialog were needed to guess all of them correctly. For another 15 concepts that were already in the semantic memory and have not been yet corrected by the active learning process about 2.4 failed games were needed to guess them all correctly. Certainly the knowledge acquired in an automatic way may help to boostrap the process, but extension and verification of this knowledge is at least equally important.

To evaluate the efficiency of active learning approach to knowledge acquisition a detailed descriptions for 8 arbitrary chosen concepts have been manually created and converted to CDVs. These handcrafted prototypes served as a golden standard. Starting from the automatic procedure initial knowledge representation has been created and the active dialog used in the learning process. The CDVs for the chosen concepts became gradually quite similar to the handcrafted prototypes. The Manhattan distance between prototypes CDV_{prot} and CDVs stored in semantic memory CDV_{sm} gives an approximate measure of the concept description error that is reduced due to learning. The total error is calculated according to the formula:

$$E_{avg} = \sum_{k}^{N_p} |CDV_{kprot} - CDV_{ksm}|/N_p \qquad (6)$$

where Np is the number of prototypes used in training. The change of this error is shown in Fig. 1 as a function of C_{dlg}, the number of dialogs that were finished with the semantic memory actualization. Initially a lot of knowledge has been missing but the error decreases slowly as a function of the number of dialogs in an approximately linear way. It may not decrease much further as the knowledge added through the active dialogues is not aimed at reducing this difference and achieving perfect description of concepts. Minimum amount of knowledge that enables to play the twenty question game successfully will remove the problems and stop invoking the dialogues. Thus only knowledge needed for specific purpose is acquired in this way.

The method for acquiring data using the twenty questions game quickly learns the most popular concepts people think about. In our experiments group of 30 students were asked to play the game. In the first 30 games using scenario C system learned 17 new concepts. The new 30 games played by the same group of people created 12 new concepts. When another group of 20 people was asked to play the game only 4 new concepts were created.

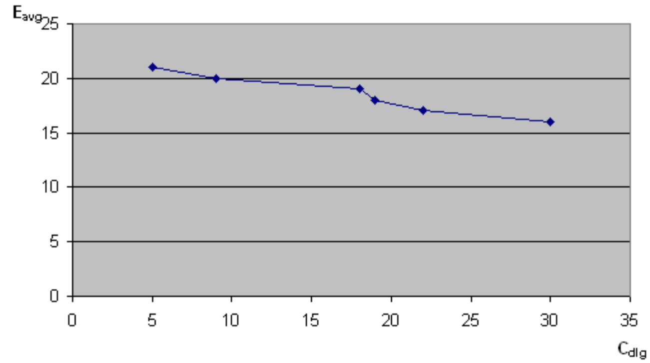

Fig. 1. Change of the average number of errors as a function of the number of games played.

The initial semantic memory created from text sources could not distinguished 15 concepts. They were all correctly separated with 6 passes of the dialog presented in section B. The system acquired 5 new keywords and 4 new types of relations.

Initially the program based on data obtained directly from machine-readable structured sources won only 5 out of 30 games. After refining the concept descriptions stored in CDV vectors through the active dialogs for 30 concepts, the number of games played for these concepts that were finished with success increased to 28, corresponding to 93,3% success rate. On average only 6.6 questions per game were needed before correct concept has been guessed. The games were played with assumption that the player gives correct answers. The average number of CDV entries (concept-keyword relations) after semantic memory has been automatically created was quite low, only 4.6 keywords per concept. After playing 90 games followed after unsuccessful games by an active dialogue this number has been increased to 11.7.

V. Discussion and future directions

Creating a good semantic memory is critical to many applications of the natural language processing (NLP). The problems addressed in this paper are somewhere between text mining and knowledge representation and acquisition. Text mining techniques have been used to create initial semantic memory [3], including active learning techniques for finding new relations and features for concepts that may be used in the 20 questions game. AI approach to NLP has been very ambitious, trying to parse and understand quite complicated sentences [14] and store knowledge in the form of complex frames. In effect this knowledge is too difficult to use it in word games of the 20 question type. Statistical approaches to language on the other hand [15] have completely abandoned structural description of concepts in favor of co-occurrence relations. In this paper we have used concept description vectors, a very simple vector representation aimed at capturing some structural properties defining natural concepts.

Automatic creation of semantic memory from ontologies, dictionaries and other structured sources unfortunately is still too difficult, therefore it needs to be corrected and extended. One way to acquire more knowledge is through collaborative projects, such as FrameNet [16]. However, collaborative projects should bootstrap themselves using knowledge that already has been accumulated in the semantic memory. Several examples of data acquisition in the active dialog have been presented. They show the opportunities for building system capable of learning through interaction with people. The system presented here is focused on nouns, not unlike the human semantic memory that is involved in capturing the basic meaning of natural and abstract objects. More complex models of human cognition may be build but even the simple models such as the one presented here can exhibit some linguistic competence.

Active dialogs presented here have only limited application as more dialog scenarios are needed. However, scenarios presented here have already helped to correct many semantic memory errors, acquire new concepts and increase the number of relations by generalization of some properties to the higher ontological level. New concepts may be added in an iterative way in the learning process based on interactions in dialog with many users, thus the system will bootstrap itself quickly if the number of such interactions will be large. The experiments performed so far were restricted to a few dialog scenarios and a narrow domain.

Due to complication of parsing complex answers the dialog with the human is limited to very simple sentences. The user answers should contain approximately only 3-4 words. This simplification of the user answers to the shortest form is enforced in the verification stage, where the program rejects sentences it can't parse and asks for answers in the simplest form. As a proof of concept hard-coded sentence templates and rules for dialog scenarios have been presented. Adding a stronger parser should allow for understanding of more complex sentence structures. Dialog scenarios connected with specified wCRK constructions eg.: `<concept> - <predicate HAS> - <noun>` are very promising. They allow for building questions in the form `why?, what for?`.

Good semantic memory is an invaluable source of information and will have quite broad applications, including natural language dialog systems. The method presented here may be expanded to more general domains and more specific dialogs may be added. Using an open architecture a system that helps to create new dialogs for knowledge acquisition may be developed and applied to word games without any restrictions. Implementation of more dialog scenarios and tests of semantic memory improvements in answering questions will be carried out in the near future. Question answering can be done using predefined question templates and identification of suitable information within semantic structures. This approach requires additional rules for data processing. The rules are connected with dedicated methods for processing relationships between concepts, for example: answering whether falcon can fly requires the use of mechanism for properties inheritance through is_a relation. Answering whether `kiwi can fly` requires the use of previous rule enriched with additional methods for ranking knowledge from different ontology levels. This relatively simple methods for text processing give a chance for turning semantic memory into a powerful knowledge base.

The program has been built as a standalone application, working on a domain-centric data, in our example from the animal kingdom. To make it more universal it is necessary to embark on a large scale project, broadening the domains and initiating a collaborative, web-based project to collect more data. One idea to scale this whole approach up is to build a set of virtual agents. Each of them will talk only in the domain of interest it specializes in. The results of interactions of these agents can be later integrated into one large system.

Building such system on a large scale will require manual implementation of many dialog scenarios that are related to various types of relations. It should be possible to add meta-level, building new types of relations and methods for processing them using frames instead of coding algorithms. The processing frames can be taken from the FrameNet [16], and adding them to sentence templates it should be possible to incorporate them into semantic memory. Building a large system as an open architecture project and providing methodology for coding dialog scenarios should be possible in a cooperative community effort. This would be an important step towards flexible natural language interfaces.

Acknowledgment

Support by the Polish Committee for Scientific Research, research grant 2005-2007, is gratefully acknowledged.

References

[1] A.M. Collins, M.R. Quillian, *Retrieval time from semantic memory.* Journal of Verbal Learning and Verbal Behavior 8, 240-7, 1969.
[2] A.M. Collins, E.F. Loftus. *A spreading-activation theory of semantic processing.* Psychological Reviews 82, 407-28, 1975.
[3] J. Szymanski, T. Sarnatowicz, W. Duch, Towards Avatars with Artificial Minds: Role of Semantic Memory. Journal of Ubiquitous Computing and Intelligence, American Scientific Publishers (in print)
[4] J.F. Sowa, ed. *Principles of Semantic Networks: Explorations in the Representation of Knowledge.* Morgan Kaufmann Publishers, San Mateo, CA, 1991.
[5] D.B. Lenat, *CYC: A Large-Scale Investment in Knowledge Infrastructure.* Comm. of the ACM 38, 33-38, 1995.
[6] UMLS Knowledge Server web site: http://umlsks.nlm.nih.gov
[7] P. Matykiewicz, W. Duch, J. Pestian, Nonambiguous Concept Mapping in Medical Domain, Lecture Notes in Artificial Intelligence, Vol. 4029 (2006) 941-950
[8] http://wordnet.princeton.edu
[9] See http://web.media.mit.edu/ hugo/conceptnet
[10] See www.ontologyportal.org
[11] A. Newell, *Unified theories of cognition,* Harvard Univ. Press, Cambridge, MA, 1990.
[12] See various word frequency lists at www.kilgarriff.co.uk
[13] http://www.natcorp.ox.ac.uk/
[14] D.B. Lenat, *CYC: A Large-Scale Investment in Knowledge Infrastructure.* Comm. of the ACM 38, 33-38, 1995.
[15] C.D. Manning and H. Schütze, *F*oundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA, 1999.
[16] http://framenet.icsi.berkeley.edu/