Semantic Memory Architecture for Knowledge Acquisition and Management

Julian Szymański
Department of Electronic, Telecommunication and Informatics
Gdańsk University of Technology, Poland
Email: julian.szymanski@eti.pg.gda.pl

Włodzisław Duch
Department of Informatics, Nicolaus Copernicus University Toruń, Poland
Google: Duch

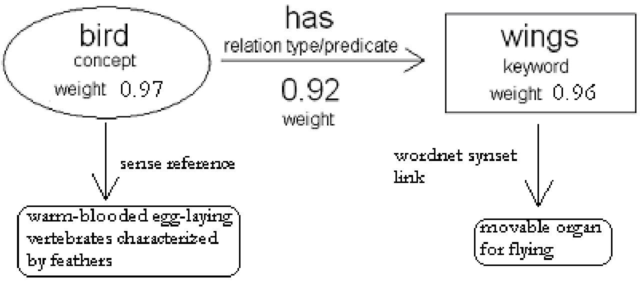
Abstract: Understanding information contained in texts and symbols requires extensive background knowledge. This knowledge is stored in semantic memory. Recently manually created ontologies and dictionaries became very popular but still do not contain sufficient information for many applications, such as semantic search or automatic annotation of concepts. Architecture facilitating knowledge acquisition for common sense semantic memories is presented and an algorithm for playing 20 questions word game used for further verification and acquisition of new knowledge. This approach is then applied to the WordNet database for gathering new and correcting existing knowledge.

Keywords: knowledge acquisition, commonsense knowledge, ontologies, semantic memory, Wordnet, text understanding, semantic Internet, NLP.

I. Introduction

Common sense knowledge may be represented in semantic network models [1] based on relations between objects and their features. Unfortunately such large-scale knowledge models supplying well defined descriptions of concepts that could be used in numerous applications do not exist. In addition existing data sources are static, without the ability to acquire new knowledge. The only system capable of understanding complex information structures contained in texts is the human brain, therefore good systems should be inspired by psycholinguistic theories of human cognition. Human understanding is a complex process involving different types of memories. One of them is a container of conceptually related data where basic meanings of concepts describing real objects are stored. According to Tulving conception of information organization in human cognition this container is called semantic memory (SM) [2]. A few SM theories exist, the most popular being Collins and Quillian hierarchical model [3], Collins and Loftus model of spreading activation [4], and the feature comparison model [5].

These psychological theories can be simulated in artificial systems using atomic logical element: a typed relation (predicate) between concept (object) and keyword (feature) with weight value expressing the strength of this relation. This structure is called here wCRK (weighed Concept - Relation Type - Keyword). It can be compared with the popular Resource Description Framework triple [6] used by the semantic internet community, with added weight value and URI (uniform resource locator) given to distinguish a particular sense of the term. An example of wCRK is shown in the picture below. The triplet: concept – predicate – keyword, forms a "word layer" of the system. According to the Chomsky's ideas about language [7], words are only indicators of deep semantic structure of language. In our system the "sense space" of words is implemented using WordNet lexical resources [8].

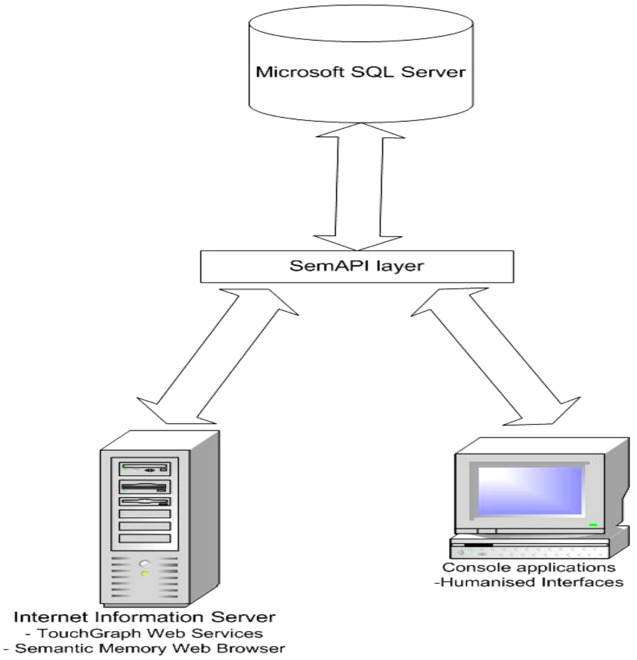


In this paper construction of semantic memory based on analysis of free text information that may be found in the internet is described. Such SM may be used in natural language dialogue systems, word games, formulation of questions, more precise queries for semantic search and many other applications [9]. Unfortunately automatic creation of good semantic memories is a very difficult task. A new approach to this problem is presented and tested below. First the architecture of the knowledge management system is described in Sec. 2, and then construction of the knowledge base, followed by results of some experiments.

II. System architecture

Semantic memory system (described below) has been built using Microsoft .NET technology. The Microsoft IIS web server is used as an application server. It assures access to the system from the heterogenic environments, WWW interfaces and web services for cooperative semantic memory usage and multimedia interfaces. The business intelligence layer between them, called semAPI, is a set of libraries encapsulating communication with database and realizing numerical calculations on the semantic space. Such approach facilitates scalability, clear functional borders and makes complex data operations easy, providing uniform database access for different applications.

Presence of numerous many-to-many relations poses high demands on system processing. A relational database offers standard and widely accepted mechanisms to deal with this issue. Microsoft SQLServer was used here as a container for conceptual information storage (Fig. 1 below). The database tables can be divided in two logical groups according to functionality they are involved in. First group ensures reference between words and their senses using WordNet mappings. The second group consist of tables for holding wCRK structures, where weighted relations allow for modifications and learning of new knowledge.


Microsoft SQL Server
SemAPI layer
Console applications
-Humanised Interfaces
Internet Information Server
- TouchGraph Web Services
- Semantic Memory Web Browser

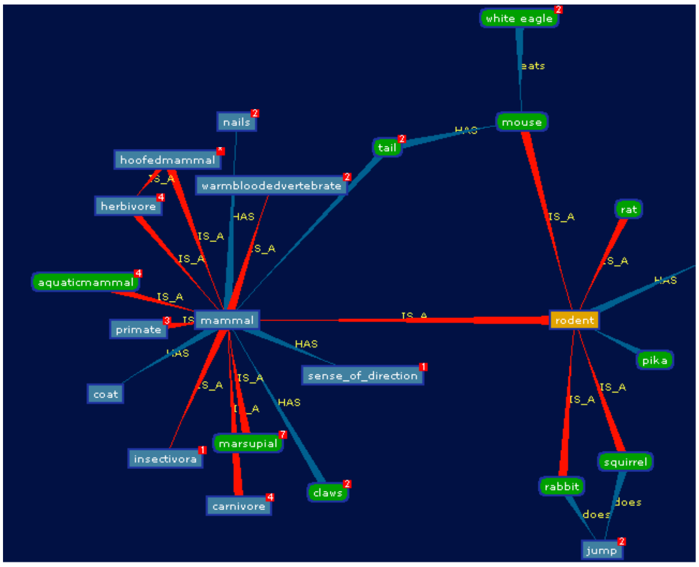
Direct operations on a database require tedious programming. To make this process as flexible as possible and programmer-friendly an intermediate translation layer was implemented. This business intelligence layer maps database tuples on the application objects making the interaction with database transparent for programmers. This layer implements classes with static methods for SQL queries affecting mapped table. Each of the tables has a corresponding class, and a data structure that allows access to the fields in the database tables. Each of the class provides methods: Insert, Update, Delete and Select for the operation on specified entity. The data tuples are automatically mapped to these structures during the object creation process. It ensures realization of atomic functions for building and operating large logical structures, such as classes for numerical calculations performed on semantic space, providing fast algorithm for turning ontolgy-oriented semantic representations to their numerical representations (see below).

III. Interfaces

The semantic memory system can process data through many interfaces. They serve as gateways for users to retrieve or enter new data.

a. Semantic space visualization

Data stored in semantic memory in wCRK form are hard to analyze directly. To make this work easier the TouchGraph [10] component has been used, providing graphical tool for visualization of contextual data. A java applet working with semantic memory web services gives an interactive graphical network of concepts, enabling an easy navigation through this space. Selecting particular node shows its details and links to related objects. The nodes and edges can be modified manually: the applet allows operations such as adding, editing and deleting components of the semantic space. The data changed in this way is marked as "manual" to distinguish the hand-crafted knowledge from learned or automatically generated knowledge. An example of such visual presentation of ontology-oriented representation of the semantic space is shown in Fig. 2 below.



b. Numerical semantic space representation

Neither the semantic wCRK structures nor the database tuples are useful for direct numerical calculations. However, the semantic space can be converted to the matrix of the Concept Description Vectors (CDV) – the numerical representation of relations for each concept [9]. The CDV vector components describe the strength of relations between particular keyword and the concept represented by the vector. This very simple knowledge representation enables numerical processing of the information contained in the semantic space. Although it may not be sufficient for full parsing and understanding of texts it is useful for many applications. For example, semantic query system should understand what the user has in mind, and if this is not clear should ask minimum number of questions to gain additional knowledge. The 20 questions game serves as a good example for such question-answer applications, requiring an algorithm for guessing concepts that the user thinks about. In the simplest case the system is asking questions and the user answers 'yes' or 'no'. Using matrix representation of concept space where rows represent M objects (o) and columns represent N features (c) the best semantic space property maximizes information gain:

$$IG(c_m) = -\sum_{i=0}^{N} p(o_i) \log p(o_i)$$

$$\text{where} \quad p(o_i) = \text{abs}(w_{mi})/N$$

where w_{mi} is the strength of relation between object o an its feature c. If the weight is positive the keyword is relevant to the concept, if it is negative it is known that the concept does not have the particular property described by the keyword, and if the weight is 0 then the keyword is not applicable to the concept at all. The status may also be undetermined, the NULL relation weight codes for the lack of data.

In each step of the game feature that will maximize the information is calculated from the subspace of the most probable concepts. The relevant subspace is built from initial query and previous answers using the formula:

$$O(A) = \| CDV, ANS \| = \min$$

where the distance ||·|| between all concept description vectors (*CDV*) describing all concepts and the vector representing user answers (*ANS*) is minimal. The 0 value in the answer vector codes user answer "don't know". The distance calculation is performed according to the formula:

$$DM(CDV, ANS) = \frac{\sum_{n-1}^{N} \left(1 - d(CDV_n, ANS_n)\right)}{\|ANS\|},$$

$$\text{where} \quad d(x,y) = \left\{ \begin{array}{lll} 0 & \mathit{for} & y = NULL \\ \dfrac{|0-y|}{2} & \mathit{for} & x = NULL \\ |x-y| & \mathit{otherwise} & \end{array} \right\}$$

where x and y represent weights for *CDV* and *ANS* vectors and ||*ANS*|| is the Euclidean length of the answer vector. This measure is used instead of previously used cosine distance [9] as it allows to calculate distance between vectors that have some undefined components. Note that for *CDV* not all weights have definite values (because of the lack of data), and the *ANS* vector is formed only from user answers, so at the beginning it has very few components with numerical values. The difference measure *DM* allows for interpretation of different ways the user answers "don't know", coded as 0 in *ANS*, and 0 in *CDV* describing as "not applicable" relation between concept and feature. Also NULL value in *CDV* can be interpreted separately.

The initial semantic space is built with only "positive" relations, describing the concept using its properties. Thus the lack of relation with some feature is interpreted in negative sense, assuming that it implies that the concept has no such feature. Due to the knowledge incompleteness this assumption is frequently false and the knowledge in SM needs to be corrected in the learning process described below.

c. Natural dialog interface

The data stored as wCRK can be used to formulate simple sentences. The sentence generator (semSentenceGenerator) module in semAPI layer enables creation of simple queries based on selected wCRK. They serve as input data for the Humanized Interface (HIT) architecture incorporating three modules: Haptek [11] talking head, text to speech synthesis, and speech recognition. This interface is used in the human – machine dialog for data acquisition (Fig. 3 below).

The items imported into semantic memory from WordNet are not directly useful in query precisiation or word games. WordNet descriptions may contain many specialized terms (for example biological taxonomy terms) that are not known to most users, while a lot of knowledge that is obvious to humans is not explicitly mentioned. To verify and complete this data a version of the 20 questions game is used and an interactive information exchange initiated with the users. Questions about particular semantic memory assertions (wCRK) are formulated using specific dialog scenarios and answers used to improve representation of concepts in the semantic space. Learning is realized through modification of the values estimating the strength of relations between concepts and features. If the assertion is true weights are incremented, for unconfirmed relations weights are weakened.

Large weight between particular feature and concept means that the answer should be useful with high certainty, as this knowledge is about something widely known. In our experiments two active dialog scenarios were implemented:

- Concepts acquisition: this is run when the SM-based system fails to guess the precise concept in the 20-question game. Using scenario: "I give up. Tell me what did you think of?" system can acquire a new concept. Representation of this concept in form of CDV vector is formed using answers obtained during the game. For exiting objects SM system can correct strengths of relations of the concept with features that appeared during the game.

- Acquisition of new features: the second scenario “Tell me what is characteristic for <concept> “ is used to separate two concepts that have very similar CDV representation. This dialog is run when the SM system fails to discern some concepts during the game. Using it iteratively for similar concepts new descriptive features are introduced to the system.

These two simple scenarios allow to collect and clean SM knowledge. The learning process based on user answers obtained during games bootstraps on the existing knowledge and is as an alternative for handcrafting ontologies. Each of the finished games (failed or succeeded) causes data actualization – correction of weights describing relations. The actual weight is calculated as a mean of all answers of users, calculated from the arbitrary point in the past (w_0):

$$w = \frac{w_0 * \beta + \sum^{N} ANS}{N + \beta}$$

where β is a certainty factor describing trust in generic w_0 value, N is the number of user answers, and ANS is the vector with answers. For $w_0 = 0$ also β=0. This approach protects SM system from random wrong answers of some users.

IV. Experiment and results

Verification and improvement of the quality of SM knowledge using the 20 questions game has been tested on the semantic space created for animal kingdom domain. Specialized biomedical or technical domains may be more impressive, but animal domain is better for tests because it is quite large and still easy to understand, as most people share similar knowledge related to animals.

Initial semantic space was generated using WordNet 2.0 database [8]. Animal kingdom concepts have been filtered using semantic category = animals. Hypernym and meronym relation types for providing descriptive features for objects from this category have been used. Initially 7 543 concept synsets (synonym sets) have been created in the domain taxonomy hierarchy and 1 696 synsets as their features. Semantic space was created with 145 227 relations, giving average CDV density of 19.25 features per object. 6 128 synsets are leaves in taxonomy, the rest are names of some animal families, such as *canine* or *insects*. It is doubtful that a typical human will know the names of so many animals. Default weights have been introduced to account for popularity of concepts. Weight values were based on word popularity taken as normalized measure of tree components and calculated according to the formula:

$$Rank(word) = \frac{\frac{IC}{\max(IC)} + \frac{GR}{\max(IC)} + \frac{BNC}{\max(BNC)}}{\max(Rank)}$$

where *IC* is the information content calculated from the probability of finding a particular word in the WordNet descriptions, *GR* is the GoogleRank (number of web pages returned by Google search engine for a given word) and *BNC* is the frequency of synset words taken from the British National Corpus [12]. Based on this measure most probable concepts with popularity index in the highest 15% of all concepts were selected. Ten concepts that have been removed in this way are: water flea, wahoo, white whale, yellow jacket, nematoda, river horse, dominick, rock bass, blastocyst, escherichia coli.

Also the 20% of the features with lowest word rank were cast off. To make generation of the sentences in the dialog simpler we assume that the most probable word in the synset is its representative. Semantic spaces based on synsets are reduced in this way to semantic spaces based on representative words. This adaptation generates space for 889 objects described with their 420 features. The average CDV density is 23.62 from 21 000 relations, so the reduced space contains better descriptions. Removing concepts that had no explicit relation to the “animal” feature reduced space to 676 most popular animals, with 329 features describing them using 19 456 relations. The average CDV density is now 28.78. Below CDV for 4 sample objects are presented:

Table I. Example of generated descriptions

Puma		Giraffe	
is_a	animal	is_a	animal
is_a	beast	is_a	beast
is_a	being	is_a	being
is_a	brute	is_a	brute
is_a	carnivore	is_a	creature
is_a	cat	is_a	entity
is_a	creature	is_a	fauna
is_a	entity	is_a	mammal
is_a	fauna	is_a	mammalian
is_a	feline	is_a	object
is_a	mammal	is_a	organism
is_a	mammalian	is_a	placental
is_a	object	is_a	vertebrate
is_a	organism	has	belly
is_a	placental	has	body part
is_a	vertebrate	has	cannon
is_a	wildcat	has	cell
has	belly	has	chest
has	body part	has	coat
has	cell	has	costa
has	chest	has	digit
has	coat	has	face
has	costa	has	hair
has	digit	has	head
has	face	has	hock
has	hair	has	hoof
has	head	has	rib
has	paw	has	shank
has	rib	has	tail

has	tail	has	thorax
has	thorax		

Cobra		Butterfly	
is_a	animal	is_a	animal
is_a	beast	is_a	arthropod
is_a	being	is_a	beast
is_a	brute	is_a	being
is_a	creature	is_a	brute
is_a	entity	is_a	creature
is_a	fauna	is_a	entity
is_a	object	is_a	fauna
is_a	organism	is_a	insect
is_a	reptile	is_a	invertebrate
is_a	serpent	is_a	object
is_a	snake	is_a	organism
is_a	vertebrate	has	ala
has	belly	has	body part
has	body part	has	cell
has	cell	has	cuticle
has	chest	has	face
has	costa	has	foot
has	digit	has	head
has	face	has	shell
has	head	has	shield
has	rib	has	thorax
has	tail	has	wing
has	thorax		

Testing the quality of the semantic space is based on 20 questions algorithm. The proportion of the failed and succeeded games gives some estimation of the quality of SM. The games give also an opportunity for learning new or correcting already existing knowledge. Below the first games won by the SM system are presented in simplified form. Despite application of statistical filters concepts that are not clearly related or are not being commonly known appear. If several user answer “don’t know” for some features they will be marked with R and removed from subsequent games.

Puma: [has chest] R, [is vertebrate] Y, [is placental] R, [is mammalian] R, [is mammal] Y, [has canon] R, [has shank] R, [has hock] R, [has hoof] N, [has paw] Y, [has pulp] R, [has root] R, [has stump] R, [has socket] R, [has matrix] R, [has corpus] R, [has marrow] R, [has crown] R, [is a canine] N, [is cat] Y, [is wildcat] Y, [is leopard] N, [is puma] Y. System correctly guess concept ‘puma’.

Cobra: [is vertebrate] Y, [is mammal] N, [has plumage] R, [is bird] N, [has flipper] N, [is reptile] Y, [is serpent] R, [is snake] Y, [is viper] N, [is boa] N, [is racer] Y, [is cobra] Y, [is asp] ASP. System correctly guess concept ‘cobra’.

Butterfly: [is vertebrate] N, [has cell] Y, [is invertebrate Y], [is insect] Y, [is butterfly] Y, [is copper] N, [is emperor] N, [is admiral] N, [is monarch] N, [is viceroy] N, [is comma] N, [is large white] N. System correctly guess ‘butterfly’.

Giraffe: [is vertebrate] Y, [is mammal] Y, [has hoof] Y, [is equine] N, [is bovine] N, [is deer] N, [is swine] N, [has horn] N, [has horn] N, [is sheep] N, [is antelope] N, [is bison] N. System correctly guess concept ‘giraffe’.

Stork: [is vertebrate] Y, [is mammal] N, [is bird] Y, [has comb] N, [is waterfowl] N, [is hawk] N, [is sandpiper] N, [is thrush] N, [is finch] N, [is parrot] N, [is starling] N, [is stork] Y, [is adjutant] N. System correctly guess concept ‘stork’.

Not all games were finished with success. Some concepts are organized in WordNet in a different way than in the ordinary human knowledge. Below examples of failed games are presented with corrections of knowledge learned from these failures:

Lion: [is vertebrate] Y, [is mammal] Y, [has hoof] N, [has paw] Y, [is canine] N, [is cat] Y, [is wildcat] Y

Organization of the lion concept in the WordNet taxonomy causes the game to go in a wrong way and the program fails:

[is leopard] N, [is panther] N, [is puma] N, [is lynx] N. I give up. What was it?

After obtaining the correct answer SM system reorganizes its knowledge and in the next game guessing the lion concept is recognized using additional *mane* feature:

[is vertebrate] Y, [is mammal] Y, [has hoof] N , [has paw] Y, [is canine] N, [is cat] Y, [is wildcat] Y, [is leopard] N, [has mane] Y. I guess it is a lion.

The second example involves searching for concept in a different branch of taxonomy - invertebrates.

Spider: [is vertebrate] N, [has wing] N, [has tail] N, [is invertebrate] Y, [has shield] N

The system knows that the spider has shield because it is anthropod, and shield is characteristic for anthropods. This question is answered by ordinary human as No, so this divergence causes the SM system to fail:

[is worm] N, [is anemone] N, [is coral] N, [is polyp] N, [is medusa] N. I give up. What was it?

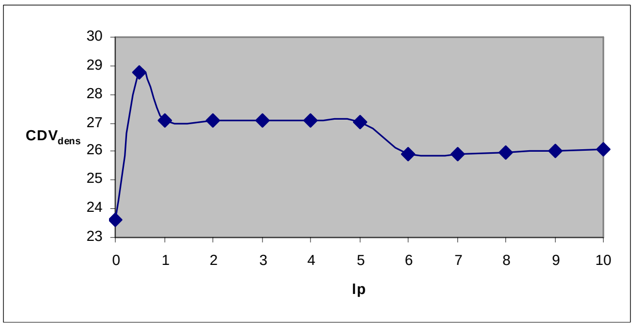
This defeat causes semantic memory reorganization and in the next game with the same concept questions are:

Spider: [is mammal] N, [has wing] N, [has tail] N, [is invertebrate] Y, [has shield] N, [is worm] N, [is spider] Y

Above examples show results of the approach for automatically generation of semantic spaces. For the first 10 concepts 78% of the games finished with success. The second measure for estimating improvements of the quality of semantic memory is the average number of games necessary to correctly learn a concept that initially has not been guessed correctly. This is estimated as the average proportion of failed games N_f performed until success is achieved. For the first ten wrongly recognized concepts it was:

$$QN = N_f / N = 2.39$$

Each game causes semantic memory actualization. The most prominent changes occurs during first games when non-relevant or not commonly known features are eliminated (decreasing part of the graph). This process rapidly reduces the average CDV density of features. The change in the density of CDV features is plotted below for initial data matching vs. the number of games. Obtaining new, verified knowledge with such a large number of concepts requires quite many games to improve SM and therefore the speed of knowledge acquisition is slow (last part of the graph).



V. Future development

The importance of ontologies in information systems will certainly grow. Lack of well defined common sense ontologies that could easily acquire new knowledge and lack of good algorithms to build them is a major obstacle in improving the natural language interfaces. This is clearly seen for hand-crafted lexical resources such as WordNet: although it has been a collective effort of many people it contains a lot of noise, and very specialized knowledge that few people understand, while information that is obvious to humans is missing (every human knows what a 'horse' is, so no-one describes it explicitly in dictionaries or encyclopedias, but 'canon' or 'shank' body parts are used, although few people know such concepts). The active learning algorithm presented here may be used for building and verification of common-sense ontologies. The approach presented here stands as an alternative for declarative approach where knowledge is manually enter to the system. Construction of semantic memories and ontologies through observation of human actions and interactions with users is a promising way to build computer programs capable of using natural language. It is also a better way to gain knowledge and learn language then declarative approaches used in such large-scale systems as CyC [13]. To gain quickly a lot of common-sense knowledge a cooperative project will be pursued, creating word games and inviting many users to play them on web pages.

One of the inconveniences of the 20 questions game algorithm presented here is the requirement of getting correct user answers (according to common sense knowledge). Despite some robustness of the learning algorithm to mistakes in user answers games in which users gives wrong answers usually fail. A better mechanism for handling user mistakes is planned by changing the selection of the most probable concepts subspaces.

The extensions of active dialogs with other scenarios should lead to demonstration of better linguistic competences in artificial systems, going beyond keyword identification in the dialog and application of sentence templates. This has already been done in the Eliza program of Weizenbaum [14]. A large-scale semantic memory should allow for real concept understanding. In the next stage we shall implement dialog scenarios for verification of the new assertions generated using analogies between wCRK.

The presented approach seems to be general and applicable to many other knowledge domains.

References

[1] Sowa, J.F. ed. Principles of Semantic Networks: Explorations in the Representation of Knowledge. Morgan Kaufmann Publishers, San Mateo, CA, 1991.

[2] Tulving E, Episodic and Semantic Memory; in: Tulving, E, Donaldson W (Eds): Organization of Memory. New York 1972.

[3] Collins A.M. and Quillian M.R, Retrieval time from semantic memory. Journal of Verbal Learning and Verbal Behavior 8, 240-7, 1969.

[4] Collins A.M. and Loftus E.F, A spreading-activation theory of semantic processing. Psychological Reviews 82, 407-28, 1975.

[5] Smith, E.E., Shoben, E.J., and Rips, L.J. Structure and process in semantic memory: A featural model for semantic decisions. Psychological Review, 81, 214 241, 1974.

[6] RDF description, see: http://www.w3.org/RDF/

[7] Chomsky N, Aspects of the Theory of Syntax. MIT Press 1965.

[8] Wordnet, see: http://wordnet.princeton.edu/

[9] Szymanski J, Sarnatowicz T, Duch W, Towards Avatars with Artificial Minds: Role of Semantic Memory. Journal of Ubiquitous Computing and Intelligence, American Scientific Publishers (in print)

[10] Touchgraph, see: http://www.touchgraph.com/

[11] Haptek avatars, see: http://www.haptek.com

[12] BNC, available at: http://www.natcorp.ox.ac.uk/

[13] Lenat D.B, CYC: A Large-Scale Investment in Knowl-edge Infrastructure. Comm. of the ACM 38, 33-38, 1995.

[14] Weizenbaum J, Computer Power and Human Reason: From Judgment to Calculation. W. H. Freeman & Co. New York, USA 1976.

Julian Szymański received the BEng and MSc degrees from Gdańsk University of Technology in computer science and from the Nicolaus Copernicus University in philosophy; he works currently as the teaching assistant at Gdańsk University of Technology.

Wlodzislaw Duch received the MSc, PhD and DSc degree from the Nicolaus Copernicus University, and is the Head of Department of Informatics at this university; recently he has been Visiting Professor at Nanyang Technological University in 2003-2007, and currently serves as the President of the European Neural Networks Society.
For more information Google: W. Duch.