Towards Avatars with Artificial Minds: Role of Semantic Memory

Włodzisław Duch,[†] Julian Szymański,[‡] and Tomasz Sarnatowicz[§]
School of Computer Engineering, Nanyang Technological University, Singapore

The first step towards creating avatars with human-like artificial minds is to give them human-like memory structures with an access to general knowledge about the world. This type of knowledge is stored in semantic memory. Although many approaches to modeling of semantic memories have been proposed they are not very useful in real life applications because they lack knowledge comparable to the common sense that humans have, and they cannot be implemented in a computationally efficient way. The most drastic simplification of semantic memory leading to the simplest knowledge representation that is sufficient for many applications is based on the Concept Description Vectors (CDVs) that store, for each concept, an information whether a given property is applicable to this concept or not. Unfortunately even such simple information about real objects or concepts is not available. Experiments with automatic creation of concept description vectors from various sources, including ontologies, dictionaries, encyclopedias and unstructured text sources are described. Haptek-based talking head that has an access to this memory has been created as an example of a humanized interface (HIT) that can interact with web pages and exchange information in a natural way. A few examples of applications of an avatar with semantic memory are given, including the twenty questions game and automatic creation of word puzzles.

I. INTRODUCTION

A lot of efforts in constructing interfaces based on natural language processing (NLP) have been devoted to cheating the user that the program understands the meaning of the words. Since the famous "Eliza" program of Weizenbaum [1] chatterbots attempt to discover keywords and sustain dialog by asking pre-prepared questions without understanding the subject of conversation or the meaning of individual words. This is quite evident from the Loebner prize chatterbot competition [2], popularity of bots based on AIML language [3], and the slow progress in text understanding and natural language dialogue systems. Although AIML provides tens of thousands of templates to associate input statements with linguistic outputs cheating has obviously its limitations and it is doubtful that good natural language interfaces may ever be built this way. Humans may use cheating from time to time to hide their ignorance, but most of the time they understand concepts used in the linguistic constructions, have rich associations that enable formulation of reasonable questions, and have in mind concept representation and associations that go well beyond what may be captured in templates.

The use of language with its grammatical structure is specific only to humans, but even pre-linguistic associations and understanding of natural objects by animals have not yet been captured in artificial systems. True understanding of language concepts requires various types of memory systems to facilitate concept and object recognition, description and building episodic relations among concepts during discourse or text analysis. This would not be possible without rich semantic information about the world, ability to recall object properties, concepts, relations and possible actions stored in the associative memory. Information processing in mammals is dominated by vision, creating the difficulty of reducing the knowledge that animals and humans have about natural world to linguistic description. Representation of the words in the brain includes phonological subnetwork pointing to the series of microcircuits that resonate when phonemes that form the sound are received from the auditory stream, linked to microcircuits that code graphemes for visual recognition of the word form, and to extended subnetworks that involve, visual, auditory, sensory and motor cortices [4][5]. Understanding what is being observed and experienced engages the perception-action networks. Transition probabilities between activations of such subnetworks reflect similarity and associations between concepts. So far NLP techniques have not drawn inspirations from neurocognitive understanding of processes that brains use to understand and create language.

[†]also at Department of Informatics, Nicolaus Copernicus University, Toruń, Poland; Google: Duch
[‡]Department of Electronic Telecommunication and Informatics, Gdańsk University of Technology, Poland;
Email julian.szymanski@eti.pg.gda.pl
[§]Email: tomasz.sarnatowicz@globalintech.com.pl

Some attempts to capture relations between concepts have been focused on ontologies and semantic memories. Although the properties of semantic memory [6] may be partially captured by semantic networks [7]-[10] so far this technique has been successfully applied only in narrow domains [14], and it is not easy to see how to create a large-scale semantic network that could for example be used to answer questions or sustain a meaningful dialog with a chatterbot. Of course other types of memory: recognition, procedural, episodic and working memories are needed to create artificial mind endowed with the inner world, but the knowledge contained in the semantic memory is the foundation upon which other cognitive elements of artificial minds should be built. Although the idea of semantic networks is relatively old and there are many projects that label themselves as "semantic networks", it is surprising that rich description of simple natural objects, such as animals, plants, body organs or body states (disease), have never been constructed. Large-scale projects such as Wordnet [15] contain a single-sentence descriptions of natural objects, and relations of the synonym, hypernym-hyponym (superordinate/subordinate), antonym, entailment, and meronym/holonym type. There is no simple way to find out that cats have teeth, claws, tail, make different sound and drink milk. Without such information that easily comes to mind of anyone who knows what a cat is, creation of artificial minds or even good dialog systems does not seem possible. CYC [16] has been the most ambitious project based on sophisticated frame-based knowledge representation. It has been pursued for decades and can potentially be useful in natural language processing, although this has yet to be demonstrated. However, CYC does not provide simple associations, it has no mechanism to invoke properties of objects that are needed in a discourse. The complexity of the knowledge-based reasoning in large systems makes it unsuitable for real-time tasks, such as quick analysis of large amounts of text found on the web pages, or simultaneous interactions with many users.

In this paper cognitive inspirations are drawn upon to make the first step towards creation of avatars equipped with semantic memory that will be able to use language in an intelligent way. This requires the ability to ask questions relevant to the subject of discourse, questions that constrain and narrow down possible ambiguities. Previous attempts to use semantic memory in natural language processing systems have been too ambitious, trying to encode not only object properties but also many types of relations [9–11]. After many years of development of the FrameNet project [12, 13] only 792 frames has been defined (as of April 2006). Although in principle such semantic memory may be used as a basis for language understanding in practice progress is slow and complex knowledge representation schemes have not proved to be really useful so far.

A different strategy is followed here: in many applications very simple knowledge representation for semantic memory is sufficient, therefore one should investigate the potential and the limits of the simplest approaches first. In the next section concept description vectors (CDV) are introduced, providing the basic for rudimentary semantic memory. Drawing on such semantic memory an avatar may formulate and may answer many questions that would require exponentially large number of templates in AIML [3] or other such languages.

Our goal is to create Humanized InTerface (HIT) based on a 3D human head model, with speech synthesis and recognition, which could be used to interact with programs (on Web pages or local software), making the interaction much more natural than typing. Endowing avatars with linguistic abilities involves two major tasks: building semantic memory model, and providing all necessary means for natural communication. In the next section talking head based on the Haptek avatar is described. Actions of the avatar should be based primarily on the knowledge stored in its semantic memory. Building such memory is not a simple task and requires development of automatic and manual data collection and retrieval algorithms, using many tools for analysis of natural language sources. Issues related to knowledge representation and creation of semantic memory are described in section 3, a few sample applications in section 4, and remarks on extending the semantic memory in section 5. Discussion and future directions close this paper.

II. HAPTEK TALKING HEAD

Haptek [20] provides tools for building 3-D avatars that may be added to web pages (Haptek player is installed as a plugin in Netscape, Internet Explorer or Mozilla browsers under Windows), or used as an embedded component in custom programs. Haptek's *PeoplePutty* tools have been used (commercial, but inexpensive) to create a talking head (full-body characters capable of gestures may also be created). This package includes tools for custom modeling, morphing faces, adding accessories (such as hats or glasses), building custom gestures, adding textures or external 3D rendering environment backgrounds, and using 3rd party animation systems (for example, motion capture).

High-fidelity natural voice synthesis with lips synchronization may be added to Haptek characters. Free MS Speech Engine [21], (MS Speech API - SAPI) has been used to add text to speech synthesis and speech to text voice recognition, but other commercial or open source packages can also be used. It is also possible to play files in the open streamable audio format OGG created by Vorbis [24], which could be useful for example to sing or talk in a specific voice. Haptek movements, gestures, face expressions and animation sequences may be programmed and coordinated with speech using JavaScript, Visual Basic, Active-X Controls, C++, or ToolBook. The actual view of our talking head is shown in Fig. 1.

FIG. 1: Haptek-based talking head was used as an interface to play the 20-questions game.

This is an example of a humanized interface (HIT) that, with little need for external programming, can interact with web pages and send information both ways, hiding the details from the user. Interaction with Web pages is based on Microsoft .NET framework [25]. However, the use of an avatar for anything else then simple translation of text-to-speech and vice versa requires some understanding of language and some knowledge about the world. For that reason semantic memory has been created as a basis for future avatar mind.

III. SEMANTIC MEMORY

Human understanding of the world would be impossible without semantic memory [6], storing structured representations of knowledge about the world entities, concepts and relations between them. Semantic memory is a permanent storage of conceptual data. "Permanent" means that data is collected throughout the whole lifetime of the system, even though old information can be overridden or corrected by newer input. "Conceptual" means that this type of memory contains semantic relations between words and uses them to create concept definitions. Semantic memory in practical applications should be a container for storage, efficient retrieval and information mining. Two approaches have been used here to realize it: Collins and Quillian hierarchic model of semantic memory [7] and Collins and Loftus spreading activation model [8]. Our implementation is based on the connectionist part of this model and uses relational database and object access layer application programming interface (API).

The database stores three types of data: concepts, or objects being described, keywords (features of concepts extracted from data sources) and relations between them. Types of relations (like "x IS y", or "x CAN DO y" etc.) are defined when input data is read from dictionaries and from ontologies (at present for the free text input "IS RELATED TO" is the only relation used). The only predefined relation is the IS-A relation used to build ontology tree, which serves for spreading activation, i.e. features inheritance down the ontology tree. Semantic memory has been created in an automatic way using relational database that stores many types of relations between concepts.

For some applications a much simpler knowledge representation, based on Concept Description Vectors (CDVs), is sufficient. CDV store for each object an information whether a given property can be applied to this object or not. Although information about relations is lost for some applications the gain in computational efficiency is more important.

A Building semantic memory: collecting data

There are two most important goals that should be satisfied to create a useful large scale model of semantic memory. The first one is technical, an efficient implementation of the model. It was achieved by using relational database and by creating specialized data access API to operate on data stored in it.

The API serves as data access layer providing logical operations between raw data and higher application layers. Data stored in the database is mapped into application objects and the API allows for retrieving specific concepts/keywords, comparing them, checking separability of certain conditions. This gives clear data operating interface, and from the data storage side – an effective method for storage of large amounts of data.

The second goal is more difficult to achieve: the memory must be filled with appropriate data. There are two major types of data sources for semantic memory: machine-readable dictionaries that have an internal structure that allows for direct conversion into semantic memory data structures, and blocks of text, which include mainly definitions of objects from dictionaries and encyclopedias.

So far three machine-readable data sources have been used. The Suggested Upper Merged Ontology (SUMO) and its domain ontologies form the largest formal ontology in public domain, with about 20,000 terms and 60,000 axioms [17]. SUMO is the only formal ontology that has been mapped to the entire WordNet lexicon [15]. It includes the MId-Level Ontology (MILO). Sumo/Milo provided ontology tree for the semantic memory. ConceptNet [18] is a freely available commonsense

knowledgebase and natural-language-processing toolkit. It has been generated automatically from the large corpus of about 700,000 sentences collected in the Open Mind Common Sense Project, a World Wide Web based collaboration in which over 14,000 authors typed all kinds of obvious "common sense" facts. The concise ConceptNet knowledgebase has 200,000 assertions and the full base contains 1.6 million assertions. These assertions cover the spatial, physical, social, temporal, and psychological aspects of everyday life. They capture a wide range of commonsense concepts and relations in a simple, easy-to-use semantic network, like WordNet, which has been used as the third main source of data.

WordNet is the largest hand-crafted project of its kind, with more than 200,000 words-sense pairs. It may be described as "a lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets" [15]. ConceptNet is focused on concepts, while WordNet is focused more on words. ConceptNet has more diverse relational ontology then WordNet, facilitating creation of practical, context-oriented, commonsense inferences for processing of real-world texts.

Information from individual sources was loaded separately into its own workspace. Functions are provided to combine and match it for further processing. The most basic workspace used for most of further calculations is based on the IS-A relation imported from WordNet hypernymic relations (this is "a kind of " relation). To save storage and processing time in initial computational experiments objects and keywords were limited to animal kingdom only. Hyponym and meronym relations from WordNet were also added. Note that WordNet defines relations between synsets (synonym sets), not individual concepts. Other dictionaries use only words, so for compatibility all WordNet data was converted into words before storing. This enables adding this information to relations stored in ConceptNet. Relation types such as: CapableOf, PropertyOf, PartOf, MadeOf, have been imported. The ConceptNet IS-A relation and Sumo/Milo ontology served as verification for a given *a priori* WordNet hypernimic relations. The effect of this approach was enhancing factors of ontological relations and bringing up the most characteristic of them. WordNet and ConceptNet relations were then compared, and new types of relations were created, including only those pairs (concept-keyword) that were considered related in both dictionaries.

For free-text data we have used three dictionaries: Merriam-Webster [26] , WordNet [15] and Tiscali [27]. Whole word definitions were stored in memory as meanings linked to concepts, so that they could be easily accessed in future applications. They were processed using the following algorithm.

1. For each concept, based on their definitions, three sets of words have been built (one for each source dictionary).
2. Each word has been replaced with a synset (set of synonyms from Wordnet).
3. The expanded sets of words were then compared and the common part of all three has been mapped back to synsets, resulting in a set of synsets that are most likely related to the initial concept.

The effect of application of this procedure is a set of most characteristic words from definitions of a given concept. They were stored as a separate relation type. We met here a problem of articles and prepositions, and words such as 'having', 'none' etc. which at this level of analysis do not contribute any useful information. They were removed by using a manually created stop-word list. Phrases are more informative than words. To extract them out of free text blocks a tagger based on ApplePieParser [19] engine has been used. Before saving them into memory, concept-phrase relations were compared with concept-keyword ones and only the phrases that matched keywords were used. This ensured that only sensible phrases were stored in the memory.

B Concept Description Vectors

Although semantic memory contains a lot of important information some questions may be answered in a more efficient way, without numerous inferences and graph searching, using vector-based knowledge representation. For a given type of relation all keywords from semantic memory create semantic space and all concepts may be treated as points in this space. Merging all types of relations reduces them to the most general one - "x IS RELATED TO y" - which merges all different semantic subspaces into a single one. It is then natural to use a binary vector description of concepts, called here CDVs (concept description vectors). These vectors should simply indicate which properties are related or have sense for a given concept. They are similar to the context vectors for a given concept, although there is an important difference. Concepts that are found close to each other in arbitrary texts may not be related, while concepts derived by algorithms described in previous sections should be related. For most objects described by nouns CDV contain information about properties of these objects.

All CDV vectors create matrix representation of the reduced information contained in semantic memory. It is important to notice that most of concept-keyword pairs have no value. Formally, all empty cells in a semantic matrix should be considered as default value of "Unknown", or "Not Applicable" in a Concept-Keyword relation. However, from technical point of view it seems natural to treat such cells as actually empty and only use the default value at runtime whenever necessary. The

CDV matrix is very sparse, facilitating easy storage of large amounts of semantic data. The CDV representation has lost potentially important information - actual types of concept-property relations - for the sake of clarity and ease of computation. Statistical analysis using more faithful representation of semantic memory is rather difficult. Using reduced representation enables answers to many questions that would not be feasible otherwise. For example, trying to find an interesting word puzzle one may notice that there is only one concept with a subset of 3 features that are applicable to this object (CDV bits = 1), and ask a question: *what has charm, spin and charge?* Putting these keywords in Google search correctly brings the page "Quarks" at the top. The number of interesting questions that an avatar may ask using simple analysis of CDV matrix is very large. Some more applications are described in section 4.

IV. APPLICATIONS

Haptek avatars may be used as plug-ins in most WWW browsers, connecting to web pages and communicating their contents, or simply transmitting queries and reading answers from specific fields in web forms. Instead of reading and typing natural communication may be established, hiding the details of word games and other applications from the user. This approach may be used to play simple games, such as trivia games, where one has to choose among several possible answers. The number of human responses is very limited (for example, a, b or c), making the speech recognition part quite easy.

Another interesting application is to use the avatar in combination with a chatterbot. The Ai Research [22] features "The HAL Nursery", also called "the world's first Child-Machine Nursery". This is a software program using reinforcement learning techniques to acquire language, through trial and error process similar to that infants are using. Ai is hosting a collection of "Virtual Children", or HAL personalities developed by many users through mere conversation. At present the HAL software reaches the level of an 18 month old child, producing, after longer training, coherent two-three word sentences. Practice and training is done via typing and reading, but an avatar in the form of a child head and the ability to speak in a baby-like voice will make it much more interesting to play with. Our Haptek head has also been used as an interface hiding the text-based interface of another chatterbot named Allan, present at the Ai Research web page [22].

These applications do not need to take advantage of the semantic memory. Two word games have been created that rely on the CDV reduced representation: the 20 questions game and the puzzle generator. The 20 question game is very important because the ability to play it well also implies the ability to ask questions that can reduce ambiguity of any query. This has quite obvious implications to improve search results by adding additional keywords and removing some keywords from the search. The avatar has been used to play the 20 question game via the web page [23], but this version is based on pre-defined questions, not on the semantic memory.

A The 20 question game

In its basic form the goal of the 20 question game is to guess a concept that the user has in mind by asking yes-no questions [23]. In such form the game requires solution of only two algorithmic problems:
1) Construction of questions to be asked in the right order (which depends on previous answers). Due to the simplicity of CDV the questions are usually similar and have a simple form of "Is it related to X?", or "Can it be associated with X?", where X is a keyword stored in the semantic memory. The questions could be formed in much more human-like way, but for our purpose this awkward form carries sufficient information. Note that ignoring the problem of forming a question leaves only the need to select a keyword, not necessarily build the whole question. However, once the keyword has been selected it is possible to use the full power of semantic memory to analyze the type of relations and ask more sophisticated questions.
2) A scoring function that ranks the concepts that should still be considered at a given stage of the game, based on the answers received so far. The concepts with the highest score are the best candidates to be the answer. If the score of the top concept is significantly higher than that of the next concept a direct attempt to guess the answer is made.

The first of these problems has been solved by checking how much information will be gained if a given keyword is selected:

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

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

p_i is the fraction of all concepts that have not been eliminated so far and for which the keyword has value v_i and K is the number of possible relation values (in case of binary CDV vectors $K = 2$). The best strategy is to find a keyword for which approximately half of the concepts have CDV(keyword,concept)=1 and the remaining half value 0. Care should be taken to reduce influence of wrong answers that could remove the correct answer from the list of concepts currently considered. The vector of currently received answers A defines a subset of objects O(A) that are the most probable answers, with a uniform probability

$$p(A) = 1/|O(A)| \tag{3}$$

This probability could be changed if additional information is accumulated from many games about the *a priori* probabilities of different concepts. All vectors in O(A) have zero distance in the subspace spanned by A keywords. To take into account the possibility of errors in the answers a larger subset O(A+k) of concepts at a distance k from the O(A) concepts should also be taken into account with probability

$$p(A) = 1/|O(A+k)| \tag{4}$$

An extension to the basic form of the game is by admitting more possible answers - except for "Yes" and "No" the following answers are also accepted: "Unknown", "Seldom", "Sometimes" and "Not Applicable". The second problem - ranking the concepts - has been solved by calculating distance from each concept in the semantic memory to the currently received answers. If the answers are not binary Yes/No, the distance $||K-A||$ is calculated in the following way:

$$||K-A|| = \sqrt{\sum_i |K_i - A_i|^2} \tag{5}$$

where $|K_i - A_i|$ depends on the type of relation K_i and answer A_i:

- if either K_i or A_i is Unknown then $|K_i - A_i| = 0.5$
- if either K_i or A_i is Not Applicable then $|K_i - A_i| = 1$
- otherwise K_i and A_i are assigned numerical values: Yes=1, Sometimes = 2/3, Seldom = 1/3, No = 0 and $|K_i - A_i|$ is actually calculated

In our computer experiments all concepts stored in the CDV matrix were parts of a single ontology reduced to animal kingdom to avoid storage size problems. The first few steps based on binary splits found keywords with information gains close to 1, indicating that the two subsets have similar number of elements.

Unfortunately due to the difficulty of automatic creation of CDV vectors they are very sparse, with 5-20 (average 8 for the whole set) out of several thousand keywords that may have definite values. As a result of this sparseness in later stages of the game one question may lead to elimination of only very few concepts. This requires either using other methods of limiting the number of concepts or making the semantic matrix much denser. However, automatic creation of high quality semantic memory is not an easy task.

B Word Puzzle Generator

The second application that has been created using semantic memory is designed to work as a web service that invents word puzzles and uses the avatar as an interface to the game. Semantic memory is the only source of data. The application selects a random concept from all concepts in the memory and searches for a minimal set of features necessary to uniquely define it. If many subsets are sufficient for unique definition one of them is selected randomly. Formation of questions is based on simple rules, depending on the value of concept-keyword relation (which in its basic form can only be Yes/No) and on the part-of-speech tag of the keyword. Thus for nouns, verbs and adjectives the form of the question is:

- Nouns N - X is a (an) N, or X is not a (an) N.
- Verbs V - X is able to V, or X is unable to V, or X can V, or X cannot V, or X V, or X does not V.
- Adjectives A - X is A, or X is not A.

Additions of words like "but", "and", etc, allows for creation of sensible questions like:

"It is a mammal, it lives in water and it lays eggs. What is it?" (A duckbill).

"It is a bird, but it does not fly and it has a long neck. What can it be?" (An ostrich).

Some more sample puzzles generated in the program are:

"It is a rodent, it is intelligent it has fur and a long tail. What do you think it is?" (A rat).

"It is an Amphibian, it is orange and has black spots. How do you call this animal?" (A Salamander).

Although using the current knowledge base the answers to these questions are unique, the player thinking about an animal that has not been included may find more than one correct answer. This gives an opportunity to expand the knowledge base, although addition of new information may require its verification.

In this application only a very simple forms of phrases were used that seemed sufficient at this stage of the project. Semantic memory stored in the CDV reduced knowledge representation allows for great flexibility, with almost infinite number of questions that may be invented. Similar approach may be used in educational applications, testing knowledge in selected fields.

V. EXTENDING SEMANTIC MEMORY DATA

Linguistic competence of systems using Semantic Memory depends greatly on amount and quality of information stored in the memory. Initial set of data was created from several available dictionaries that could be processed automatically. They do not, however, contain full information necessary for more complex human-computer conversation. Therefore it is necessary to extend the semantic network with new nodes (features and concepts) and relations between them. This task can be performed in two ways: by querying humans in a conversation, or by automatic extraction from text corpora. Following two subsections describe both approaches.

A Active search

To enrich concepts descriptions stored in semantic memory automatic data acquisition from a text corpus written in a natural language is attempted. WordNet dictionary definitions (glosses) were used as the source of information. It appears that the glosses contain some information that can be transformed into relations. WordNet itself stores relation-based knowledge, but in most cases the original relations are far from being complete (i.e. fully cover actual human knowledge). To extract new relations from the free-text descriptions (glosses) an algorithm called Active Search was created. Its major component is a rule-based criterion function which decides whether a given feature applies to a concept, defined as:
FC(*feat*, *def*(C)) = **TRUE** if *def* defines *feat* as a feature of the concept C it describes, **FALSE** if *def* determines that *feat* is not a feature of C and **NA** if either of above cannot be determined.

At first all meronym relations stored in WordNet were extended. Using WordNet as the source of information there is no need to extract subject of a free text block - it is given *a priori*. To simplify the algorithm it is assumed that everything in the definition refers to the main concept. To avoid limitations enforced by computational complexity and/or resources the search was limited to the body parts in definitions of animals. The set of potential body parts (keywords) was established as a hyponyms of the "body part" concept in WordNet (over 100 names), and the set of search definitions came from glosses of hyponyms of the "animal" concept. The algorithm of searching animal body parts is realized in the following steps:

1. Choose a keyword K and find the set of definitions DEF(K) containing K as a plain word. Note that definitions come with concepts related to them, so DEF(K) is also a set of concepts.

2. Run the criterion function against each definition in DEF(K) to decide whether K actually applies to a concept the definition describes. This step requires an application of a set of predefined rules:

   (a) Suffix analysis:
      i. Find W_K, the actual word that contains K.
      ii. Find B_{WK}, the basic form of W_K.
      iii. If W_K is a concatenation of B_{WK} and one of the known suffixes, return TRUE or FALSE depending on what type of suffix was encountered. Otherwise, proceed to step (b).
      The first two steps should remove cases where K is a part of some other word, but has no actual semantic relation to it. For example if K="horn", it can be found in a word "thorn".

   (b) Analysis of surrounding words
      i. Find W_K, the actual word that contains K.
      ii. Find B_{WK}, basic form of W_K.
      iii. If $B_{WK} = K$, search its neighborhood for one of known verbs or modifying words (see below). If found, return TRUE or FALSE, depending on what words in what forms were found.

Relation of an animal and its body parts can be expressed in many ways. Currently the search is limited to forms similar to "X has Y" (using "have", "have got", "posses" etc.) and to the word "with", which is also often used in definitions. These are indicators of the positive value of the relation (i.e. result in returning TRUE from the function). Similarly, we consider forms of "have not" or "without" as indicators of the negative relation between the animal and a body part (return FALSE). Successful classification examples of the algorithm are:

F_C("tail")=TRUE, according to suffix analysis: {catamount, lynx} "short-**tail*ed***" wildcats with usually tufted ears; valued for their fur"

F_C("tail")=FALSE, according to suffix analysis: {anuran, batrachian, frog, salientian, toad, toad frog} "any of various **tail*less*** stout-bodied amphibians with long hind limbs for leaping; semi-aquatic and terrestrial species"

F_C("tail")=TRUE, according to surrounding words:
{fox} "alert carnivorous mammal ***with*** pointed muzzle and ears and a bushy **tail**; most are predators that do not hunt in packs"

{coney, cony, rabbit} "any of various burrowing animals of the family Leporidae ***having*** long ears and short **tails**; some domesticated and raised for pets or food"

F_C("horn")=FALSE, according to surrounding words {psittacosaur, psittacosaurus} "primitive dinosaur actually ***lacking*** **horns** and having only the beginning of a frill; long hind and short front limbs; may have been bipedal"

Some results were obtained wrongly. For example:

F_C("hair")=FALSE, according to suffix analysis. {mouse} "any of numerous small rodents typically resembling diminutive rats having pointed snouts and small ears on elongated bodies with slender usually **hair*less*** tails"

The whole mouse is of course not hairless. This case should be correctly solved with a deeper sentence analysis, which would limit the scope of the word "hairless" to its noun phrase only. Active search enriched the set of meronymic relations by about 10%. There were initially 569508 such relations (defined directly and distributed to lower nodes of animal taxonomy). The algorithm found 5773 new relations, which after distribution in the taxonomy expanded to a number of 53534. Initial relations

were formed by 2103 concepts and 546 features. Active search found relations with 267 keywords, of which only 65 were in the memory before. In other words, Active Search algorithm found 202 body parts that were not initially related to animals. WordNet stores only positive values of meronymic relations. The algorithm found 104 negative relations with 55 features.

B Interactive completion of the semantic memory data

The contents of semantic memory, compared to the human knowledge, is incomplete and inaccurate. Incompleteness comes from the lack of information in, and inaccurate results from faulty algorithms for knowledge extraction from text corpora, or human errors when the semantic memory content is extended in a human-computer interaction. There is always need to complete, extend, verify and confirm the SM data.

Semantic Memory in general, and ontology trees or Concept Description Vectors contain only a limited representation of knowledge. Technically speaking, two major tasks in every application that utilizes SM are storing and retrieving, listening (or reading) and analyzing user statements and composing responses. Because information in SM is neither complete nor accurate an "uncertainty" factor may be added to help in evaluation of its usefulness. An intelligent module is needed to extend and improve reliability of information stored in SM, formulating and asking questions during conversation. Below a method to add new data is presented, with verification of reliability of existing entries planned for future. Before discussing the actual generation of questions recall what knowledge is available as the input data for the generator. Semantic Memory discussed here stores two major types of information:

- relations between concepts (involving two synsets, or a concept-feature pairs);
- hierarchical ontology tree.

While the first type of data includes multiple types of relations, the second one (ontology) is actually a relation of IS-A type. This means that the whole SM is a set of more or less verified facts. The basic model with yes/no binary entries is usually too simple, it needs at least the third entry "unknown", to allow distinction between relations that have been verified as false and those that have not been verified at all. To distinguish all concepts stored in the semantic memory the values of those unknown relations that could be used for discrimination should be determined first. Several situations may arise, listed below:

1. **Situation:** there is an entry in SM that potentially relates X to Y. However, there is no value of the relation (true-false or anything in between).

   Reaction: simply ask about the particular relation.

2. **Situation**: there is a concept in SM with very limited information about it (or even no information at all).

   Reaction:

   - ask for a feature that is (positively or negatively) related to X (i.e. "Tell me something about X")
   - (if there is some ontology similarity to another concept Y)
     – ask for a common feature of X and Y (the feature may or may not exist in the SM)
     – ask for a difference between X and Y (as above)
   - ask for a feature that distinguishes X from everything else. Note that such a feature may or may not exist.

3. **Situation:** A concept X has two (or more) known features, which distinguish it from all other known concepts.

   Reaction: ask for another concept with the same set of features.

4. **Situation:** Two concepts X1, X2 are identical (have the same set of relations and share a parent in the ontology tree).

   Reaction:

   - ask for a feature that distinguishes both concepts best (new one or one with unknown value)
   - ask for another concept that is similar to the two

5. **Situation:** Two concepts X1, X2 that have some features in common and some that differ.

   Reaction:

   - (feature Z is unknown for one or both concepts) ask if Z has the same value for both concepts
   - ask for a new feature (or one with unknown value) that has identical value for both concepts
   - ask for a new feature that distinguishes both concepts best

6. **Situation:** Two concepts X1 and X2 differ by all known values.

   Reaction:

   - ask for a common feature of the two

- (if X1 and X2 share a parent in the ontology tree) ask for the most common feature of the whole branch, then ask for its values for X1 and X2

7. **Situation:** all children of X in the ontology tree have the same value of relation with feature Y, except some that are unknown. **Reaction:** ask whether Y has the same value for all children of X

Situations described above are of course just examples. Note that all rules that apply to children and parents in the ontology tree can be applied at any level of the tree, not necessarily the lowest one. All facts that are collected that apply to a concept in the middle of the tree should be propagated down to its leaves.

The best way to extract knowledge possessed by humans is to ask relevant questions. While this article does not cover all problems with actual text generation/analysis, it is worthwhile to mention possible scenarios when the above situations may occur. At the end of the 20 questions game there are always three possible situations: the program either wins, or loses by giving an incorrect answer, or ends up in a situation when no answer can be given. All three situations are a good start for a conversation. The last case is the most comfortable one. Having no possible answer means that at some point (in the final stages of a game) there were a number of concepts that could not be distinguished using the feature values in the SM. This is an opportunity to extend the set of known features or their relations with concepts (start with Situation 4 above). If the program gives a wrong answer, and user provides the correct one, there is a whole set of features that can be verified (those that led to the wrong answer) and another set of features to be added (the correct answer and the questions asked). If the program wins, there is always a chance to ask "Do you know anything else that would be a correct answer here?"

A more interesting case would be to switch roles with the user: "Tell me please, what should have I asked to obtain a correct answer". This should leads to a set of new features that will extend the semantic memory. Each modification that is done during such a conversation creates another situation which gives a chance to lead the conversation for a long time.

VI. DISCUSSION AND FUTURE DIRECTIONS

The long-term goal of our project is to create an avatar that will be able to use natural language in a meaningful way. An avatar based on the Haptek head has been created, equipped with semantic memory, and used as an interface to interactive web pages and software programs implementing word games. Many technologies have to be combined to achieve this. Such avatars may be used as a humanized interfaces for natural communication with chatterbots that may be placed in virtual environments.

Several novel ideas have been developed and presented here to achieve this:

1) Knowledge representation schemes should be optimized for different tasks. While general semantic memories have been around for some time [9, 10] they have not led to large-scale NLP applications. For some applications much simpler representation afforded by the concept description vectors is sufficient and computationally much more efficient.

2) CDV representation facilitates some applications, such as generation of word puzzles, while graph representation of general semantic memory cannot be easily used to identify a subset of features that should be used in a question. Binary CDV representation may be systematically extended towards frame-based knowledge representation, but it would be better to use multiple knowledge representation schemes optimized for different applications. Although reduction of all relations stored in semantic memory to the binary CDV matrix is a drastic simplification some applications benefit from the ability of quick evaluation of the information content in concept/feature subspaces. Careful analysis is needed to find the simplest representations that facilitate efficient analysis for different applications. Computational problems due to a very large number of keywords and concepts with the number of relations growing into millions are not serious if sparse semantic matrix representation is used.

3) An attempt to create semantic memory in an automatic way, using definitions and assertions from Wordnet and ConceptNet dictionaries, Sumo/Milo ontologies and other information sources has been made. Although analysis of that information was helpful, creating full description even for simple concepts, such as finding all properties of animals, proved to be difficult. Only a small number of relevant features have been found, despite the large sizes of databases analyzed.

Thus we have identified an important challenge for lexicographers: creation of a full description of basic concepts. Information found in dictionaries is especially brief and without extensive prior knowledge it would not be possible to learn much from them. The quality of the data retrieval (search) depends strongly on the quality of the data itself. Despite using machine readable dictionaries with verification based on dictionary glosses still spurious information may appear, for example strange relations between keywords and concepts which do not appear in real world. This happens in our semantic memory in about 20% of all entries and is much more common if context vectors are generated by parsing general texts. In most cases it is still possible to retrieve sensible data from such semantic memory. One way to reduce this effect is to parse texts using phrases and concepts rather than single words. The quality of semantic memory increases gradually as new dictionaries and other linguistic resources are added. It is also designed to be fine-tuned

during its usage. All inconsistencies will show up for example as the program mistakes in word games, giving opportunity to automatically correct them. In many applications knowing a list all features that can be applied to a given concept would be very useful, but such linguistic resources do not yet exist.

4) The active search algorithm used here with WordNet glosses may also be used with other dictionaries, encyclopedias, ConceptNet assertions or specialized sources. A good test of this approach will be to check how complete description may be inferred for some simple objects, such as animals or common objects. Unfortunately this is not a simple task, as full evaluation of the recall and retrieval rates requires good reference data, and such data may be created only with large manual effort. No-one has tried to objectively evaluate quality of a chatterbot or a word puzzle generator. An indirect evaluation of semantic memory may be done using word games.

Medical resources are especially rich. Unified Medical Language System (UMLS) [32] is a collection of many different ontologies combined by specialists, with millions of entries and about 15 million relations between its concepts. Such large number of relations comes from statistical co-occurrences, but still many associations that every medical doctor knows are not present there. This makes it hard to disambiguate clinical texts. Active learning about medical concepts, for example such body organ as a liver or heart, from medical textbooks and ontologies, should be very interesting. Creation of an avatar with similar associations as medical expert would be a major step towards intelligent medical text processing.

5) It is quite probable that automatic creation of such resources will prove to be too hard and a lot of manual effort will have to be devoted to improve the results (as was the case in such large projects as Wordnet [15]). The subject of automatically generating object description, although very fundamental, is still not popular and we have identified only a few previous attempts [28] [29] [30] have been made, but did not resulted in practical systems. In the longer run neurocognitive approaches, based on understanding language processing capabilities of human brain, may help to solve this problem [4] [5] [31].

An interesting possibility, mentioned in this paper in context of the 20 question game, is to collect relevant knowledge in a large-scale collaborative project, in the ConceptNet style [18]. The program should ask actively questions and try to bring the discussion to a higher level of ontology to gain general knowledge, not just common sense assertions. Our semantic memory may be used for bootstrapping such project, for example, if someone mentions an animal ontology is used to conclude that it is a mammal and a question "Do all mammals share this property?" is asked.

6) The 20 question game is a great test to increase precision of questions in search queries. This game may serve as the next important step on the road to pass the Turing test. An international competition in word games could be an interesting event. So far the only other program to play the 20 question game that we know of [23] is based on a weighted question/object table, that is a form of cheating, not a real knowledge about concepts. The algorithm introduced here is the only alternative so far, but to be successful in a large-scale unrestricted game it needs a good semantic memory.

In our opinion further progress in the natural language processing (NLP) requires better large-scale models of semantic memory. Without quick access to semantic memory information NLP systems will never have sufficient prior knowledge to reach high level of linguistic competence. Creating such memory, even in its simplest form based on Concept Description Vectors, is an important challenge. One of the most important tasks is to combine the full power of semantic memory with the efficiency of reduced CDV representation, and take advantage of all types of relations in all modules of the system – both in interface with humans (understanding and creating sensible sentences) and in internal data processing.

Combination of semantic memory with Haptek-based avatars may find many interesting applications. Most chatterbots try to change the topic of conversation constantly as they easily get lost in conversation. A chatterbot used with avatar equipped with semantic memory may ask intelligent questions knowing the properties of objects mentioned in the dialog. A long term goal of this research is to endow avatars with artificial mind, using cognitive architecture that will include recognition and episodic memory models as well as some reasoning ability. This is a very difficult task, but the basic problems have already been identified and some solutions proposed.

A Acknowledgement

WD is grateful for the support by the Polish Committee for Scientific Research, research grant 2005-2007.

REFERENCES

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

[2] See www.loebner.net/Prizef/loebner-prize.html

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

[4] F. Pulvermller, *The Neuroscience of Language. On Brain Circuits of Words and Serial Order.* Cambridge Uni. Press, 2003.

[5] F. Pulvermller, Y. Shtyrov, R. Ilmoniemi, *Brain signatures of meaning access in action word recognition.* Journal of Cognitive Neuroscience 17(6), 884-892, 2005.

[6] J.R. Anderson, *Learning and Memory.* J. Wiley and Sons, NY 1995.

[7] A.M. Collins, M.R. Quillian, *Retrieval time from semantic memory.* Journal of Verbal Learning and Verbal Behavior 8, 240-7, 1969.
[8] A.M. Collins, E.F. Loftus. *A spreading-activation theory of semantic processing.* Psychological Reviews 82, 407-28, 1975.
[9] J.F. Sowa, ed. *Principles of Semantic Networks: Explorations in the Representation of Knowledge.* Morgan Kaufmann Publishers, San Mateo, CA, 1991.
[10] F. Lehmann, ed. *Semantic Networks in Artificial Intelligence.* Pergamon Press, Oxford, 1992.
[11] HowNet, see www.keenage.com
[12] FrameNet, see http://framenet.icsi.berkeley.edu
[13] C.F. Baker, C.J. Fillmore and B. Cronin, *The Structure of the Framenet Database*, International Journal of Lexicography, Volume 16.3: 281-296, 2003.
[14] H. Brandt-Pook, G.A. Fink, B. Hildebrandt, F. Kummert, and G. Sagerer, *A Robust Dialogue System for Making an Appointment.* In: Int. Conf. on Spoken Language Processing, Vol. 2, pp. 693-696, Philadelphia, PA, USA, 1996.
[15] See http://wordnet.princeton.edu
[16] D.B. Lenat, *CYC: A Large-Scale Investment in Knowledge Infrastructure.* Comm. of the ACM 38, 33-38, 1995.
[17] See www.ontologyportal.org
[18] See http://web.media.mit.edu/ hugo/conceptnet
[19] See http://nlp.cs.nyu.edu/app/
[20] See www.haptek.com
[21] See www.microsoft.com/speech/default.mspx
[22] See www.a-i.com
[23] See www.20q.net
[24] See www.xiph.org/ogg/vorbis/
[25] See www.microsoft.com/net/
[26] See www.m-w.com
[27] See www.tiscali.co.uk/reference/dictionaries/animals
[28] W.B. Dolan, L. Vanderwende, and S. Richardson. *Automatically Deriving Structured Knowledge Base from On-line Dictionaries.* In: Proc. of the Pacific Association for Computational Linguistics, Vancouver, BC, 1993.
[29] L. Vanderwende, *The Analysis of Noun Sequences using Semantic Information Extracted from On-Line Dictionaries.* Ph.D. thesis, 312 p, Georgetown University, 1995.
[30] S. Richardson, *Determining Similarity and Inferring Relations in a Lexical Knowledge Base.* Ph.D. thesis, 187 p, The City University of New York, 1997.
[31] G. Hickok, and D. Poeppel, *Dorsal and ventral streams: A new framework for understanding aspects of the functional anatomy of language.* Cognition, 92, 67-99, 2004.
[32] See http://umlsks.nlm.nih.gov, the UMLS Knowledge Server web site.