Semantic Memory for Avatars in Cyberspace

Julian Szymański[1], Tomasz Sarnatowicz[1] and Włodzisław Duch[1,2]
[1]*Department of Informatics, Nicolaus Copernicus University, Toruń, Poland*
[2]*School of Computer Engineering, Nanyang Technological University, Singapore*
Google: Duch

Abstract

Avatars that show intelligent behavior should have an access to general knowledge about the world, knowledge that humans store in their semantic memories. The simplest knowledge representation for semantic memory is based on the Concept Description Vectors (CDVs) that store, for each concept, an information whether a given property can be applied to this concept or not. Unfortunately large-scale semantic memories are 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.

1. Introduction

A lot of efforts in constructing interfaces based on natural language 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 general lack of progress in text understanding and natural language dialogue systems. Cheating has obviously its limitations and it is doubtful that good natural language interfaces may be built this way. An alternative approach used by humans requires various types of memory systems to facilitate concept recognition, building episodic relations among concepts, and storing the basic information about the world, descriptions of objects, concepts, relations and possible actions in the associative semantic memory. Although the properties of semantic memory may be partially captured by semantic networks so far this has been demonstrated only in narrow domains [4], and it is not easy to see how to create a large-scale semantic network that could be used in an unrestricted dialog with a chatterbot.

In this paper cognitive inspirations are drawn upon to make a first step towards creation of avatars equipped with semantic memory that will be able to use language in an intelligent way. This requires ability to ask questions relevant to the subject of discourse, questions that constrain and narrow down possible ambiguities. Very ambitious projects, such as CYC [5], that use a sophisticated frame-based knowledge representation, have been pursued for decades and can potentially become useful in natural language processing, although this has yet to be demonstrated. However, the complexity of the knowledge-based reasoning in large systems make them 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. An alternative strategy followed here is to start from the simplest knowledge representation for semantic memory and to find applications where such representation is sufficient. Drawing on its semantic memory an avatar may formulate and may answer many questions that would require exponentially large number of templates in AIML or other such languages.

Endowing avatars with linguistic abilities involves two major tasks: building semantic memory model, and providing all necessary means for natural communication. This paper describes our attempts to create Humanized InTerface (HIT) based on a 3D human head model, with speech synthesis and recognition, which is used to interact with Web pages and local programs, making the interaction much more natural than typing.

HIT actions are based primarily on the information 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 various tools for analysis of natural language sources. In the next section issues related to knowledge representation and creation of semantic memory are described. Some remarks about Haptek avatar are in section 3, and a few sample applications in section 4. Discussion and future directions are presented in the last section.

2. 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 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 activation spreading, 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 used. 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.

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.

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 [9]. SUMO is the only formal ontology that has been mapped to the entire WordNet lexicon [10]. It includes the MId-Level Ontology (MILO). Sumo/Milo provided ontology tree for the semantic memory. ConceptNet [12] 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" [10]. 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, WordNet and Tiscali. 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 [11] 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.

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.

3. Haptek talking head

Haptek 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 [18], i.e. MS Speech API (SAPI 4 or 5) 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 [21], 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.

Haptek-based talking head has an access to semantic memory that has been created. 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 [22].

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

4. 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 [19] 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 [19].

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 [20], but this version is based on pre-defined questions, not on the semantic memory.

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. 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:

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

$$p_i = p(\text{keyword} = v_i)$$

where 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 only 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 $p(A) = 1/|O(A)|$. 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 $p(A) = 1/|O(A+k)|$.

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:

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

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.

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.

Creation of the question phrases is very simple and mechanical, but seems sufficient at this stage of the project. In near future we plan to use Markov chains to generate even more human-like sentences. Semantic memory 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.

5. Discussion and future directions

An avatar based on Haptek head has been equipped with semantic memory and used as an interface to chatterbots, interactive web pages and software programs implementing word games. This avatar may be used as humanized interface for natural communication with text-based web pages, or placed in virtual environments in cyberspace. Semantic memory stored in relational database is used efficiently in many applications after reduction to a sparse CDV matrix. A chatterbot used with avatar equipped with semantic memory may ask intelligent questions knowing the properties of objects mentioned in the dialog. Most chatterbots try to change the topic of conversation as they do get lost in the conversation.

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. In our opinion further progress in 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. Several approaches were used here to create semantic memory using definitions and assertions from Wordnet and ConceptNet dictionaries, Sumo/Milo ontologies and other information sources. 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.

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. 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).

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.

Some possibilities remain still to be explored. One of the most promising approaches is based on an active search of keywords for a given concept in ConceptNet assertions, dictionaries and encyclopedias. A good test of this approach will be to check how complete description may be inferred for some simple objects, such as animals. Another possibility is to collect relevant knowledge in a collaborative, ConceptNet style, but asking actively questions and bringing them to a higher level of ontology to gain general knowledge. 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.

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. A long term project is to control an avatar using complex cognitive architecture that will include recognition and episodic memory models as well as reasoning ability.

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] 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.

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

[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] See www.ontologyportal.org

[10] See http://wordnet.princeton.edu

[11] See http://nlp.cs.nyu.edu/app/

[12] See http://web.media.mit.edu/~hugo/conceptnet

[13] 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.

[14] L. Vanderwende, The Analysis of Noun Sequences using Semantic Information Extracted from On-Line Dictionaries. Ph.D. thesis, 312 p, Georgetown University, 1995.

[15] S. Richardson, Determining Similarity and Inferring Relations in a Lexical Knowledge Base. Ph.D. thesis, 187 p, The City University of New York, 1997.

[16] 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.

[17] See www.haptek.com

[18] See www.microsoft.com/speech/default.mspx

[19] See www.a-i.com

[20] See www.20q.net

[21] See www.xiph.org/ogg/vorbis/

[22] See www.microsoft.com/net/

[23] See web.media.mit.edu/~hugo/conceptnet/

[24] See www.m-w.com

[25] See www.tiscali.co.uk/reference/dictionaries/animals