Concept description vectors and the 20 question game

Włodzisław Duch[1,2], Julian Szymański[1], and Tomasz Sarnatowicz[1]

[1] Department of Informatics, Nicholaus Copernicus University, Toruń, Poland
[2] School of Computer Engineering, Nanyang Technological University, Singapore

Abstract. Knowledge of properties that are applicable to a given object is a necessary prerequisite to formulate intelligent question. Concept description vectors provide simplest representation of this knowledge, storing for each object information about the values of its properties. Experiments with automatic creation of concept description vectors from various sources, including ontologies, dictionaries, encyclopedias and unstructured text sources, are described. Information collected in this way is used to formulate questions that have high discriminating power in the twenty questions game.

1 Introduction

Since the famous "Eliza" program of Weizenbaum [1] chatterbot programs attempt to discover keywords and sustain dialog by asking pre-prepared questions without understanding the subject of conversation. This is quite evident from the Loebner prize chatterbot competition [2] and the lack of progress in text understanding and natural language dialogue systems. Programs that are based on semantic networks work well only in narrow domains [3].

One of the basic problems that remain unsolved is the poverty of representation of symbols. Thinking about an object – an elephant, a car, or a rose, for example – we can imagine and immediately describe general and specific properties of that object, discuss these properties and create new instances of such objects, by inventing unusual properties, for example a "pink elephant". Artificial natural language processing (NLP) systems have to make a lot of inferences to determine that a car has cylinders, while for humans interested in sports cars question "how many cylinders does it have" is quite natural. Ontologies are quite helpful to organize knowledge in hierarchical way, but it is very difficult to use ontologies to generate object description in terms of all properties it has and find which properties are not applicable to an object. Semantic networks may create a faithful representation of an episode, capturing relations in a story, but people ask questions that are outside the scope of the story because they can imagine objects and actors in more details that semantic network model is able to provide. Reasoning with rich prior knowledge may be shallow and associative, while reasoning with symbols found in ontologies and semantic networks has to be deep, and therefore difficult to perform. Tradeoffs between richness of concept representation, efficiency of use, and the depth of reasoning needed to understand questions deserve careful exploration.

Meaning of the words is reflected in their use and in the similarity of the concepts words refer to. Thinking about an object creates a set of activations of various brain modules, facilitating associations and building of semantic relations [4]. Seeing a large cat we do not need to reason that all large cats are carnivorous, hunt and may be dangerous, we immediately know it. The simplest approach to add more prior knowledge to NLP systems is to replace linguistic symbols by Concept Description Vectors (CDV). These vectors should contain information about all properties that make sense for a given concept and may be associated with it. They should contain prior knowledge that humans use to understand text, not only context knowledge that may be derived from text analysis. Although vector representation cannot do justice to all language symbols, not even to nouns, it is instructive to see how far one can go in this direction.

For the purpose of this article discussion will be restricted to CDVs for common and proper nouns only. A dictionary of concepts, and CDVs associated with them, may form a basis for intelligent selection of questions that maximizes the information gained after each answer. In this paper we state the problem, present attempts to solve it, and show some potential applications of this approach. In the next section the Concept Description Vector idea is discussed, followed by a section describing algorithms used to derive CDVs in an automatic way from text corpora. An application of these ideas to the 20 question game is discussed in section four.

2 Concept Description Vectors

Context vectors are a popular way to disambiguate word senses and capture information contained in local relations between word pairs. They contain statistical information about word co-occurrences derived from text corpora. Concept Description Vectors provide more systematic representation of prior knowledge. Previous attempts to build lexical knowledgebases from dictionaries were focused on structures of semantic relations [5,6] and analysis of noun sequences [7]. Our goal is much simpler: finding in the list of all dictionary entries those that may characterize a given concept. To achieve this goal sophisticated frame-based representations are not necessary.

How detailed should the CDV representation be? Complex objects cannot be represented in all possible details, for example CDV for some animal should not include details about all known proteins that may be found in cells of this animal. Some ontological hierarchy must be introduced. Properties that are essential to define the meaning of words appear in dictionaries and encyclopedias and should obviously be included. Unfortunately these definitions do not contain exhaustive descriptions; reading that a horse is a “solid-hoofed herbivorous quadruped domesticated since prehistoric times” (Wordnet definition [8]) does not tell us much about the horse. It is doubtful that anyone who has not seen the horse will form a reasonable idea what a horse really is, and how does it differ from a cow that may be described in identical terms. Wordnet [8] definition of a cow, a "mature female of mammals of which the male is called bull" is even less informative.

Explanation of a new concept involves description, or a set of words related to this concept. Such a description will be called a gloss. Keywords in the gloss should explain the essence of the concept, facilitating discrimination of this particular concept from all others. Dictionaries provide short definitions saturated with keywords, while encyclopedias use richer, self-explanatory descriptions. Every concept has its unique set of keywords that distinguishes it from all other concepts. Perfect synonyms should have the same set of keywords because they represent the same concept. Concepts and keywords used for their description are collected in two sets, called $\mathcal{C}$ and $\mathcal{K}$, respectively. Many words will appear in both sets, allowing for recursive expansion of descriptions. Concept Description Vectors $c(k)$ may contain numbers reflecting strength of relation between a concept and particular keyword – it can be a binary number (applicable/irrelevant), a tertiary number (yes/no/irrelevant), a small set of nominal values, a real number, an interval, or a fuzzy number. CDVs are collected in $\mathbf{S}(c, k)$ matrix containing $|\mathcal{C}|$ rows and $|\mathcal{K}|$ columns.

Binary $\mathbf{S}(c, k)$ values may be ambiguous but from computational point of view they are the easiest to handle. Real-valued matrix elements increase the expressive power of vector representation at the expense of storage and computational power needed to process such matrices. The property "color" is inapplicable to the concept of "electron", therefore the element $\mathbf{S}$(electron,color) =0. Horses have color, therefore $\mathbf{S}$(horse,color) = 1, but in the binary representation more information, such as $\mathbf{S}$(horse,color-blue) = 0 is needed, meaning that there are no blue horses. Thus negative answer to the question "can it have a blue color?" in the 20 question game will immediately eliminate horses from the list of potential animals. Binary-valued CDVs are simple but require more specific keywords than CDVs with fuzzy or nominal subset values. If some property, like "color", is irrelevant for electron, than also more specific properties like "color-blue" should be irrelevant.

The number of concepts and keywords may easily reach 10^5, making creation of CDV matrix quite a challenge. The matrix $\mathbf{S}(c, k)$ is obviously quite sparse, with most entries equal to "irrelevant". In the binary representation each concept is defined by a subset of hypercube vertices that is relatively close to the "undefined" concept (vector with zeros). In the ternary representation if the value "irrelevant" is coded as 0 then almost all concepts will lie on the centers of hypercube walls relatively close to the $\mathbf{0}$ vertex.

3 Semi-automatic creation of CDVs

Initially all keywords for CDVs have indefinite value $\mathbf{S}(c,k) = 0$. Dictionaries are the first source of reliable information, but information contained even in the best linguistic resources, such as the Wordnet electronic dictionary [8], manually created over a period of many years, is very poor. Nevertheless we have tried to use several on-line dictionaries, performing the following procedure for each entry found there:

1. create a unique list of all words used in the concept's description (gloss), ignoring duplicates and order of words;
2. convert every word to its base form, eliminate duplicates;
3. filter out all words other than nouns, verbs, adjectives and adverbs;
4. eliminate common words using stop-list (words like "be, have, thing");
5. use remaining words as indices to build CDV vectors.

Processing multiple dictionaries gives several options while creating the CDV vectors:

1. using only keywords that appeared in every dictionary;
2. using keywords from all dictionaries and merging results by a bit-wise sum;
3. using keywords from all dictionaries and creating final vectors as weighted sum of all individual vectors.

An identical procedure was used to create semantic vectors from encyclopedias. Glosses in encyclopedias usually contain full sentences, often complex ones. When analyzed, such sentences often contain "subconcepts" with their individual mini-glosses. There are also parts of a gloss which do not describe the concept directly, giving for example historical background. This makes encyclopedia-based vectors less adequate for direct creation of concept descriptions. On the other hand, using larger blocks of text reduces the risk of missing important keywords.

WordNet (ver. 2.0) and a number of other electronic dictionaries were analyzed: The American Heritage Dictionary of the English Language (4th Ed, 2000), and the Dictionary of Idioms (1997), Easton's 1897 Bible Dictionary, Merriam-Webster Medical Dictionary (2002), Webster's Revised Unabridged Dictionary (1998) and the New Millennium Dictionary of English (2003). Electronic edition of Encyclopedia Britannica (2004) was also used in some queries to expand word definitions.

CDVs were initially generated from WordNet and restricted to words used for description of animals to speed up computational experiments. Attempts to improve CDVs using context derived from larger corpora, such as a collection of books, were not successful, leading to a large number of meaningless correlations. Vectors may contain a small number of keywords that were assigned to identical set of concepts (eg. "hoofed" and "hoofed mammal").

Such duplicate vectors corresponding to perfect synonyms were removed immediately after the whole set was created, reducing the size of CDV matrix. Obvious features that apply to all concepts, and void features that are not applicable to any concepts, should be removed. Vectors created from dictionaries and free text blocks never contain void features and seldom contain obvious features, however this step should not be ignored. Another simple filter was applied to remove all keywords that were shorter than 3 characters, or appeared only in the WordNet dictionary; appearance in at least one more dictionary was a required condition confirming importance of the keyword.

Selected information about the amount of data generated for the "animal" domain by this procedure is presented below.

- Initial size of the CDV matrix was 7591 concepts and 15301 keywords
- Initial filtration reduced the size to 3995 concepts and 7108 keywords, leaving CDV matrix with about 28 million elements.
- Keywords gathered from WordNet glosses fill 0.11% of **S** matrix.
- These keywords propagated down the ontology tree fill 0.72% of **S**.
- Ontology nodes for words propagated down the tree fill 0.25% of **S**.
- Meronym relations ("has part" and "is made from") fill 0.34% of **S**.

Altogether these algorithms assign values to slightly more than 1% of the **S** matrix elements. So far learning CDVs from information contained in books and other large blocks of text proved to be difficult. Some improvements that will be explored include: 1) recognition of the parts of speech and filtering only nouns, verbs, adjectives and adverbs (as done for dictionaries); 2) analyzing noun phrases to discover concepts rather than using single words; 3) filtering weak associations; 4) bootstrapping from simple to complex concepts.

Glosses generated from dictionaries or blocks of free text are a "descriptive" source of semantic information. Another important source useful to create CDVs is derived from ontologies that contain hierarchical relations between pairs of words. Finding that one concept is a member of more general category, for example finding that a dog is a mammal, a set of concepts defined by the concept of mammal may be inherited from this higher-level concept. In the ontology tree each node (except for root) has only one parent, so ontology itself is not a good source of information. Assigning just a parent node as a single feature would not be sufficient. Parent nodes of every concept are propagated down the ontology tree (note that concepts become features or keywords in this way - e.g. "mammal" is a concept, but "being a mammal" is a feature of all its direct and indirect ontology children). Furthermore, we can propagate also features gathered from concept glosses (e.g. "vertebrate", "skin", "milk" which are features of "mammal" itself, will also be assigned to "canine" and "dog").

Wordnet project [8] provides one of the most extensive ontologies. Besides defining an ontology it includes also several other types of relations. Two types of relations have been used here: the hierarchical relation called in

Wordnet “hypernym vs. hyponym”, and the meronym relation, “has a part” vs. “is made from”. These two relations have been used to build separate sets of semantic vectors. Below two sample sets of features obtained for the concept “buffalo” are presented. Four groups of keywords are listed, obtained from several sources. In Wordnet “buffalo” is defined as: “large shaggy-haired brown bison of North American plains”.

1. dictionary glosses
   wrong: north
   correct: brown, large, shaggy, bison, plain, american
2. gloss keywords propagated down the ontology tree
   wrong: each, similar, enclose, even, less, number, north, various, young, small, column, relate, compartment, man, subclass, hollow, divide, voluntary, characterize, functional, three, short, four, bear, except, several, marsupial, monotreme
   correct: warm, toe, brown, segment, skeleton, blood, alive, skin, skull, stomach, spinal, shaggy, foot, brain, cranium, bison, nourish, chordata, cartilaginous, notochord, milk, mammal, hair, hump, large, placenta, head, phylum, movement, organism, hoof, bovid, cover, ruminant, cud, chew, bony, live, horn
3. ontology headers
   correct: eutherian, bison, bison bison, craniate, ruminant, chordate, bovid, ungulate, brute, mammal, artiodactyl
4. meronyms
   correct: coat, belly, vertebrate foot, pedal extremity, pectus, dactyl, psalterium, caudal appendage, fourth stomach, first stomach, abomasum, cannon, digit, caput, animal tissue, thorax, face, shank, hock, hoof, tail, chest, head, hair, pilus, pelage, third stomach, second stomach, rumen, reticulum, omasum, costa, rib

A lot of useful information has been collected here, although some of it is contradictory and confusing (ex. “large” and “small”). Adjectives should be kept with nouns, ex. “(bison,large)”, or “(bison,color-brown)”, and the same goes for numbers. For some applications rarely used words known only to experts in some fields (“artiodactyl” or “monotreme”) may be omitted. Most of the wrongly recognized keywords begin to make sense when properly grouped in phrases (e.g. north + american + plains).

4 20 question game

The original motivation for creation of CDVs came from the need to find optimal questions in the popular 20 questions game. In this game one person thinks about a word and another person has to guess this word asking no more than 20 questions. Answers should be just yes or no, although in some variants of the game a selection from a small subset of answers is allowed.

The first question that is being asked is usually: “Is this an animal, plant, mineral, or other?”

This game is interesting for several reasons. Answering queries by search engines requires the ability to ask questions that resolve ambiguities. In the 20 question game nothing is initially known, but after obtaining answers to several questions the definition of the subject becomes more precise, eg. “it is an animal, it is big, it is carnivorous”, and an imprecise query may be formed. The search system facing imprecise query may either return a lot of irrelevant material or ask additional questions. The ability to ask relevant questions is basic to any dialog. The best question should reduce ambiguity and lead to the maximum information gain, dividing the space of relevant concepts into two subspaces containing approximately the same number of concepts. Turing test is still beyond the reach of computer dialog systems, but a program that would ask interesting questions and guess what one has in mind at the human level of competence should be called “intelligent”. It is not clear how such competence would result from computing power alone, as in the case of chess or other board games. Thus the 20 question game may be presented as a challenge to the artificial intelligence community, an intermediate step on the way to the full Turing test.

The present approach to the 20 question games [9] is based on predefined questions, with some elements of learning to determine how important the question is. A matrix of objects times questions is defined, initially with some values entered manually, and zero values representing unknown importance of questions for a given object. The program is placed in the Internet and learns from each new play what is the answers to a specific question, increasing the weight for (object, question) element if the player gave the expected answer, or decreasing it if the answer was different than expected. This approach is inflexible, relying on predefined questions, similar to the chatterbot guessing answers that fits to a template, without explicit semantic representation of concepts.

The algorithm based on concept description vectors selects the best possible question in the following way. Initially nothing is known, and for each keyword the information gain has to be calculated: assuming k discrete values and $P(kw = v_i)$ being the fraction of concepts for which the keyword kw has value v_i, the information in this probability distribution is $I(kw) = -\sum_{i=1}^{k} P(kw = v_i) \log P(kw = v_i)$. The vector of currently received answers A defines a subset of concepts $O(A)$ that are the most probable answers, with a uniform prior probability distribution $P(A) = 1/|O(A)|$. This distribution may be changed if additional information is accumulated from many games about the *a priori* probability of selecting various concepts. All vectors in $O(A)$ subset 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 may also be taken into account, with smaller probability $P(A) = 1/|O(A_{+k})|$. Selecting next keyword that maximizes $I(kw)$ a question that has a simple form is asked: "Is it related to ...", or "Can it be associated with ...". Questions could be formed in much more human-like way, but for our purpose this awkward form carries sufficient information.

In our computer experiments all concepts stored in the CDV matrix were parts of a single ontology restricted to animal kingdom to simplify the experiments. The first few steps based on binary splits give information gain close to 1, indicating that the two subsets have similar number of elements. Unfortunately CDV vectors created automatically are very sparse, with only 5-20 definite values (on average 8 throughout the whole set) out of several thousand keywords. As a result in the later stages of the game, in the reduced $O(A)$ subspaces, each answer to a question may eliminate only a few concepts. This requires either using other methods of limiting the number of concepts or improving information in the CDVs.

Three algorithms for the 20-question game have been implemented. The first one is based on the algorithm described above and is the simplest of the three. If there are keywords that have definite values for at least half of the concepts (are applicable to these concepts) in $O(A)$ subset choose the keyword that has the largest information index. Sample game is presented below. Answer and $I(kw)$ are given in parenthesis for each keyword used in the question; the concept "buffalo" was discovered after 12 questions.

- wing (0.635)[**NO**], coat (0.680)[**YES**]
- carnivore (0.623)[**NO**], hoof (0.674)[**YES**]
- ruminant (0.692)[**YES**]
- withers (0.566)[**NO**], bovine (0.629)[**NO**], antelope (0.679)[**NO**], goat (0.538)[**NO**], bovid (0.555)[**YES**]
- wild sheep (0.628)[**NO**], buffalo (0.681)[**OK**]

If the $\mathbf{S}(c,k)$ matrix in the $O(A)$ subspace has too few definite elements, the second algorithm is used, based on the most common feature. Choose the keyword that has definite value in the largest number of concepts, and reduce the subspace of candidate concepts depending on the answer for this keyword. Because even the most common feature is assigned to a small number of concepts only, this can be either a good or a bad choice. This methods implicitly defines a prior in favor of common concepts. Most frequent keyword are usually associated with the most common concepts, so if the user has chosen a common word rather than a rare word this is a good approach. An important fact here is that void and obvious features are not present in the matrix $\mathbf{S}(c,k)$. Filtering has been done initially for the whole matrix but it should be repeated after each reduction of the $O(A)$ subspace during the game. Sample game is presented below, won in 15 questions:

- throax (0.326)[**YES**], belly (0.441)[**YES**], coat (0.666)[**YES**], eutherian (0.178)[**YES**], carnivore (0.625)[**NO**]
- hoof (0.676)[**YES**], artiodactyl (0.673)[**YES**], ruminant (0.375)[**YES**], bovid (0.505)[**YES**], bovine (0.619)[**NO**]

- antelope (0.665)[**NO**], sheep (0.603)[**NO**], goat (0.595)[**NO**], wild sheep (0.628)[**NO**], buffalo (0.681)[**OK**]

Third algorithm is based on an associative matrix memory implemented by a neural network without hidden layers [10]. The matrix $\mathbf{S}(c, k)$ is treated here as a binary weight matrix that codes the existence of association between keywords $K(k)$ (inputs, row vectors) and concepts $C(c)$ (outputs). The main steps of the algorithm are:

1. Set all elements of C to 1 (all concepts are equally probable).
2. Calculate $K = C \cdot \mathbf{S}$ (keywords strength in concepts).
3. Find maximal value element Key=$\max_k K(k)$.
4. Ask the question about the keyword Key.
5. Set $K(\text{Key}) = 1$ or -1, depending on the yes or no answer.
6. Calculate $C = \mathbf{S} \cdot K$
7. Repeat steps $2 - 5$ until maximal element of Co indicates the answer.

The key in this algorithm is step 2. Here it is just a result of the vector times matrix product, but it can be replaced with other ways of choosing next keyword for query. K vector stores history of the answers and its values can be modified had the user made a mistake. Unfortunately we have no space here to analyze performance of all these algorithms here, they are presented only as an illustration of the usefulness of CDV representation.

5 Conclusions and plans

In this paper an important challenge has been stated: creating concept description vectors from analysis of information in dictionaries, text corpora and ontologies. Without such information NLP systems will never have sufficient prior knowledge to reach high level of linguistic competence. Several approaches were used to create CDVs using Wordnet dictionaries, ontologies and other information sources. Inferring even the simplest description, with CDV feature values that indicate which keywords may be applied to a given concept, proved to be difficult.

The 20 question game has been presented here as a next important step on the road to pass the Turing test and as a great test to increase precision of questions. Three algorithms based on CDVs have been presented for selection of the most informative questions. The quality of these algorithms in real games depends critically on the quality of CDVs. In collaboration with the Department of Technology in Education, Nicholaus Copernicus University (Torun, Poland), experiments are being conducted to determine human competence in the 20 question game and benchmark our algorithms against people in real games.

Several new ideas to improve the 20 question game algorithms are worth exploring. Similarity between CDV vectors may be used to define semantic space areas of high concept density. Centers of such areas could represent the whole sets of concepts in the first steps of the algorithm and used as a single object with set of features common to all individual objects in the area, reducing the number of concepts/keywords to be processed. This may be achieved using either clusterization techniques or dimensionality reduction techniques based on latent semantic analysis. Performing principal component analysis for large matrices (ca. 3000 concepts and 10000 features) is computationally intensive. However, using the fact that the CDV matrices are very sparse (with only about 1% of non-zero values) an algorithm that performs all necessary calculations within minutes on an average PC may be formulated.

Further experiments along these lines are in progress. So far all large NLP projects, such as creation of Wordnet databases, relied heavily on human labor. Although our results on automatic creation of CDVs may be useful for some applications a lot of human corrections may be needed to create knowledge-rich lexical databases that are essential for the progress in many NLP subfields.

References

1. J. Weizenbaum, Computer Power and Human Reason: From Judgment to Calculation. W. H. Freeman & Co. New York, NY, USA 1976.
2. See transcripts at: http://www.loebner.net/Prizef/loebner-prize.html
3. 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.
4. G. Hickok, D. Poeppel, Dorsal and ventral streams: A new framework for understanding aspects of the functional anatomy of language. Cognition, 92: 67-99, 2004.
5. W.B. Dolan, L. Vanderwende and S. Richardson, Automatically Deriving Structured Knowledge Base from On-line Dictionaries. PACLING 93, Pacific Assoc. for Computational Linguistics, pp. 5-14, 1993.
6. S. Richardson, Determining Similarity and Inferring Relations in a Lexical Knowledge Base. Ph.D. thesis, 187 p, The City University of New York, 1997.
7. L. Vanderwende, The Analysis of Noun Sequences using Semantic Information Extracted from On-Line Dictionaries. Ph.D. thesis, 312 p, Georgetown University, 1995.
8. C. Fellbaum (Ed), WordNet. An Electronic Lexical Database. MIT Press, 1998.
9. See www.20q.net
10. T. Kohonen, *Self-Organizing Maps.* Springer-Verlag, Heidelberg Berlin, 1995.