Finding Semantic Associations on Express Lane

Vivi Năstase[*]**, Rada Mihalcea**[**]

[*]School of Computer Science
University of Ottawa
Ottawa, ON, Canada
vnastase@site.uottawa.ca

[**]Department of Computer Science and Engineering
University of North Texas
Dallas, TX, USA
rada@cs.unt.edu

Abstract

We introduce a new codification scheme for efficient computation of measures in semantic networks. The scheme is particularly useful for fast computation of semantic associations between words and implementation of an informational retrieval operator for efficient search in semantic spaces. Other applications may also be possible.

1. Introduction

Research in natural language processing focuses more and more on developing robust technologies, to be deployed on unseen texts and produce summaries, answer questions, or perform other tasks. Such approaches rely on lexical resources to extract information about word senses, and relate different senses to each other. Lexical resources that can support robust working paradigms must have broad coverage in terms of vocabulary, and a high degree of connectivity. From the point of view of the systems that use these resources, bigger is better. The broader the coverage the resource provides, the higher the chances that the words encountered in unseen texts are found in the resource. The higher the degree of connectivity, the better the chances of finding connections between words.

When large and highly interconnected electronic lexical resources have become available, like *WordNet*, *Roget's Thesaurus*, *Longman's Dictionary of Contemporary English*, the systems using them encountered a computational problem. The same attributes that make a resource desirable, size and connectivity, have a negative influence on processing times. It is computationally more expensive to find a node in a larger resource, and to find paths between nodes in a resource with a large number of connections.

In order to be able to have the cake and eat it too, we must find an appropriate encoding for lexical resources that can reduce computation times, while allowing the resources to grow. We propose such an encoding in Section 3.. We show in Section 4. that computations of various measures in the encoded resource become practically instantaneous and these codes have important practical applications in information retrieval. Moreover, computation times will not scale up with the size of the resource, as it happens with usual algorithms which require traversal of the resource. While the encoding we propose can be implemented on any resource with a Directed Acyclic Graph (DAG) structure, in this paper we focus our case study on *WordNet*.

Dedicated to the memory of Prof. Ivan Rival.

2. Using Resources

WordNet (Miller, 1995) is one of the most frequently used resources in the Natural Language Processing (NLP) community, for a variety of tasks – word-sense disambiguation (WSD), information retrieval (IR), question answering, and others. The most exploited links from the ones *WordNet* provides are the hyponym-hypernym (IS-A) links. They are used for the computation of various semantic distance metrics (see (Budanitsky and Hirst, 2001) for an overview of measures for semantic distance) for word sense disambiguation tasks, for generalization/specialization of concepts for information retrieval (Mihalcea, 2002), question answering, and others.

The most commonly used metrics computed using *WordNet* are depth, distance between synsets and most specific common subsumer. The problems in computation appear because of multiple inheritance, which will cause a system to traverse at least part of the network. For computation of depth, a system will have to find all possible paths to the top-most level and find the shortest one. To find whether two synsets are connected, and to find a path that connects them, one must explore many links in the network.

Previous encodings of *WordNet* have been inspired by the Dewey Decimal code scheme used by librarians (Mihalcea, 2002). While this encoding facilitates the computation of depth and the use of semantic wildcards for IR, it does not handle multiple inheritance in a computationally advantageous manner.

3. WordNet as an Ordered Set

Focusing on the IS-A links between *WordNet* synsets gives us a view of *WordNet* as an ordered set. An ordered set is a set of elements (in our case synsets), and a partial order – a reflexive, antisymmetric and transitive relation, which needs not hold for every pair of elements in the set (Rival, 1996). Properties and theories of ordered sets are mostly explored with mathematical tools. The idea for encoding an order set comes from Chibbani, 2003), who proposes a codification scheme to facilitate the computational exploration of issues specific to ordered sets.

Our purpose is to provide an efficient encoding of the IS-A structure of *WordNet*. The encoding process is driven by two goals: each synset should be uniquely identified, and the code for each synset should give us information about the entire structure above the synset, with which the synset it connected.

According to these goals, each synset will be assigned a code which consists of the combined codes of its parents and a unique identifier. The unique identifier will allow us to find easily all synsets which have as an ancestor a given synset. Combining the codes of the parent nodes will give each synset a code that abstracts the entire *WordNet* structure above it, i.e. that covers all synsets which are ancestors of the current synset. In this way, the code assigned to a synset contains the entire chain of information from a general concept down to a specific concept.

We perform the encoding in layers, starting with the most general concepts (the supremums[1] in our ordered set), and then assign a code to a child synset using the codes of the parent synsets and a unique number (represented in base 36 – 10 digits and 26 letters – to obtain a more compressed representation). For example[2]:
{ *person, individual* } has two parents:
{ *life form, organism* } and { *causal agent, cause* }

Both these last synsets have the same parent: { *entity, physical thing* } . We start by encoding the most general synset:
{ *entity, physical thing* } $\rightarrow$ 1

We encode synsets on each layer, by compounding the parents' codes and adding a unique number:
{ *life form, organism* } $\rightarrow$`1.2`[3]
{ *causal agent, cause* } $\rightarrow$`1.3`
{ *person, individual* } $\rightarrow$`(1.2+1.3).4` or `1.(2+3).4`

[1] A *supremum* is an element from an ordered set such that there is no other element greater than it is, relative to the order in the set

[2] Throughout the paper, due to space limitations, only the first two concepts in any given synset are listed

[3] The dot links codes on subsequent levels in the hierarchy

Our codification scheme was designed specifically for multiple inheritance. Alternative paths corresponding to each parent are joined by a plus sign.

4. Applications

This section outlines applications of the new encoding scheme for computation of semantic associations between words, and the implementation of an IR operator for efficient search in semantic spaces. Other applications may also be possible.

4.1. Computing Measures

The purpose of the encoding we propose for *WordNet* is to allow for faster computation of measures frequently used by NLP applications that rely on semantic networks.

Computing depth. The new *WordNet* codification enables the computation of *depth* of a node within the network without effectively traversing it, but only through simple and efficient Perl functions applied on the encoding of the node itself. Alternative paths ("plus expressions") are replaced by a `min` function, stepping on a following level (dots) with a `+`, and each level (number in base 36) with `1`. For example:
{ *accessory, accoutrement* } has the code: `6.(o+1k).j3.(1g0.23u+s7).4eu.6l6`

Following the description above, the following expression is constructed (using pattern matching and substitution in Perl):

```
6. ( o+1k ).j3. ( 1g0.23u+s7 ).4eu.6l6
1+min(1, 1 ) + 1 +min( 1 + 1 , 1 ) + 1 + 1
```

This expression is practically instantaneously obtained and computed, to give us the depth of the synset. Under the new encoding, the running time of the function that computes the depth of the node becomes O(1), compared to the original running time of O(L), where L is the length of the path from the node in consideration to the top of the hierarchy.

Finding the Most Specific Common Subsumer (MSCS). The MSCS of two nodes can be simply found by identifying the last unique number that two codes share. This operation can be efficiently performed via pattern matching. For example:
{ *accessory, accoutrement* } with code `6.(o+1k).j3.(1g0.23u+s7).4eu.6l6`
{ *merchandise* } with code `6.(o+1k).j3.1g0.291`

The MSCS is identified using one matching operation as `1g0` without performing any traversal steps across the network, as with traditional encoding. The identifier `1g0` uniquely determines the corresponding synset { *commodity, goods* } .

Semantic distance. We can find the distance between two synsets in *WordNet* by finding the MSCS, and then computing the distance between each synset and the common ancestor synset in a similar manner as depth. For the synsets { *accessory, accoutrement* } and { *merchandise* } above, we have identified the MSCS as the synset identified by the code `1g0`. In order to compute the semantic distance between the synsets { *accessory, accoutrement* } and { *merchandise* }, we compute the distance from each synset to the MSCS: `23u.4eu.6l6` and `291`. This is performed in a manner similar to the computation of depth. A solution for more complex situations, in which the synsets do not share an MSCS or an alternative path which does not contain the MSCS is shorter is currently under research.

Apart from fast computation of various measures, this encoding allows us to perform structural analysis on *WordNet*. We can detect loops, or other unwanted configurations in the resource.

4.2. Semantic Wildcard

The new encoding scheme enables efficient searches in semantic spaces for IR applications.

Keywords in a query are often used with "generic" meanings and are intended as representatives for categories of objects. *Foxes eat hens* can match *Animals eat meat*. With current indexing and retrieval techniques this is not possible, unless both *animal* and *meat* are expanded with their subsumed concepts, which may become a tedious process. For this particular example, WordNet defines 7,980 concepts more specific than *animal*, and there are 199 entries that inherit from *meat*. We end up with more than 1,500,000 (7,980 x 199) queries to cover the entire range of possibilities. If boolean queries are allowed and the OR operator is available, a query with 8,179 (7,980 + 199) terms can be used. None of these solutions seems acceptable and this is why none of them have been used so far.

The *semantic wildcard* (Mihalcea, 2002), an information retrieval operator denoted with *#*, acts similarly to the lexical wildcard, but at semantic levels, enabling the retrieval of subsumed concepts. For instance, a search for *animal#* will match any concept that is of type *animal* (*dog*, *cat*, etc.), going beyond the explicit knowledge in texts.

4.2.1. Semantic-based Information Retrieval

The improved semantic based IR system used in these experiments contains the same main components as any other retrieval system.

Question/Query Processing
In this stage we perform a simple tokenization and part of speech tagging of the user question, followed by collocation identification and lemmatization. Depending on the notation employed by the user, three keyword types are identified.

1. Words with a semantic wildcard, denoted with #.
2. Words to be searched by their *WordNet* code, denoted with @ (synonymy marker).
3. Words with no special notation, to be sought in the index in their given form.

By default, we assume a # assigned to the answer type word, and no other notation for the rest of the words. All words that are denoted with # or @ are passed on to a word sense disambiguation component that solves their semantic ambiguity. If this step is skipped, the words are by default assigned sense one in *WordNet*, with reasonable precision (over 75% as measured on large sense annotated corpora). The results reported in this paper are based on an implementation that considers this second alternative.

For keyword identification, we use a heuristic-based algorithm, which selects, in this order: all proper nouns and quoted words, all nouns, all adjectives in superlative form, all numbers (cardinals).

The answer type word is also important, since it denotes the type of information sought (is it a *country*, *animal*, *fish*, etc.). We use an approach that selects the answer type as the head of the first noun phrase. If the answer detected is of a generic type, such as *person*, *location*, *organization*, we replace it with the corresponding named entity tag. Otherwise, the answer type word is assigned a # semantic wildcard. The answer type selection process is invoked only if there is no word a priori denoted with #.

The final query format consists of: words assigned with a semantic wildcard #, which are now represented by *code**; words with a synonymy marker that are replaced with their code (allowing the retrieval of synonyms in addition to the word itself); all other words are replaced with their baseform.

Document Processing
Documents are processed following similar steps to question processing. First, the text is tokenized and part of speech tagged. We have an additional component that involves named entity recognition. Next, we identify compound words, apply a disambiguation algorithm or, alternatively, assign to each word its default sense from WordNet. Finally we assign to each noun its corresponding *WordNet* code. At this stage, we also identify paragraphs and store them as one paragraph per line. This helps improving efficiency during paragraph retrieval.

Indexing and Retrieval
The indexing process is similar to traditional IR systems. A TF/IDF weight is assigned to each term. We index complex terms, including their *WordNet* codes, as well as named entity tags, when available. No additional stemming or stop-words elimination is performed. The retrieval system allows for flexible searches, including regular expressions. Based on

WordNet codes, we have the capability of using the *semantic wildcard* operator, in addition to the lexical wildcard. Example:

{ *animal* } has the code `6.1s.dc`

A search for any animal is expressed as *animal#*, and internally translated into `*.dc[+*)].#` (* matches any sequence of characters, # replaces one numerical code, the part in square parenthesis is optional.) The last number in a code uniquely identifies a synset, and it will appear in the codes of all its descendants. This will help identify the children of a node, allowing to expand *animal* to all concepts of type animal.

Data

For experiments with the semantic wildcard, we are using the *L.A. Times* collection, with more than 130,000 documents, adding up to 500MB of text. About 1,393 questions have been released during the TREC-8, TREC-9 and TREC-10 Q&A TREC evaluation, with associated relevance judgments. From these questions, we select only the *What* type of questions, as being the most ambiguous types of questions and the best candidates for the semantic wildcard operator. Subsequently, we identify those questions known to have an answer in the *L.A. Times* collection, and out of these 75 questions are randomly selected for further tests.

Evaluating Retrieval Effectiveness

IR systems are usually evaluated in terms of *precision*, *recall*, and *F-measure*. We also use the *success rate* (Woods, 1997) to measure how many questions are answered by the system. The *success rate* for a question/query is 1 if relevant documents/answers are found, 0 otherwise.

Experiments

To evaluate the effectiveness of the semantic wildcard, we are running two experiments. One where keywords are extracted as shown before, and queries are run against the *L.A. Times* index. The goal of this experiment is to simulate traditional keyword-based IR. Ranking of documents is provided by a TF/IDF weighting scheme.

A second experiment consists of a similar scheme, but this time the *semantic wildcard* operator is enabled. Moreover, a locality operator is enabled, which seeks to identify relevant paragraphs in the text, as opposed to entire documents. Example:

Question. What is the brightest star visible from Earth?
Relevant paragraph. In the year 296036 , Voyager 2 will make its closest approach to Sirius , the brightest star visible from Earth .
Comments. The query formed in this case is star# AND bright AND Earth. Only two answers are identified by the system, and the one listed above, which is the correct one, is ranked on the first position. Sirius is defined in WordNet as a star, and consequently was annotated as such in the text.

The *F-measure* obtained for the second experiment (20%) is doubled compared with the first experiment (9.2%), with increased *precision* and lower *recall*, as expected. The *success rate* is determined as 77.3% for the second experiment, compared with 66.0% obtained for basic keyword-based retrieval.

5. Conclusions

The encoding we propose for *WordNet* allows us to compute practically instantaneously various measures of interest for several NLP applications. The processing time does not scale up with the size of the resource, as is the case with traditional methods. Another interesting aspect is that this encoding allows us to explore the structure of the resource, and identify undesirable phenomena in an ontology, like loops for example. We have identified and signaled such structural problems.

An added bonus of this encoding scheme is the use of these codes for information retrieval with semantic wildcards, which would not be possible without them.

6. References

Budanitsky, A. and G. Hirst, 2001. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Proceedings of the Workshop on WordNet and Other Lexical Resources, in the North American Chapter of the Association for Computational Linguistics (NAACL-2001)*. Pittsburgh.

Chibbani, N., 2003. *Computational Aspects of Order Preserving Maps(Fixed Point property)*. Master's thesis, SITE, University of Ottawa.

Mihalcea, R., 2002. The semantic wildcard. In *Proceedings of the LREC Workshop on "Creating and Using Semantics for Information Retrieval and Filtering State of the Art and Future Research"*. Canary Islands, Spain.

Miller, G., 1995. Wordnet: A lexical database. *Communication of the ACM*, 38(11):39–41.

Rival, I., 1996. Introducing ordered sets: Lectures on ordered sets. Http://www.site.uottawa.ca/dept/algorithms/order/.

Woods, W.A., 1997. Conceptual indexing: A better way to organize knowledge. Technical Report SMLI TR-97-61, Sun Microsystems Laboratories. Available online at: *http://www.sun.com/ research/techrep/ 1997/abstract-61.html*.