A Method to Generate Large Common-Sense Knowledge Bases from Online Lexical Resources

Vasile Rus
Department of Computer Science
Institute for Intelligent Systems
University of Memphis
320 Dunn Hall
Memphis, TN, USA
email: vrus@memphis.edu

Abstract

This paper presents a general method to automatically build large knowledge bases from online lexical resources. While our experiments were limited to generate a knowledge base from WordNet, an online lexical database, the method is applicable to any type of dictionary organized around the elementary structure *lexical entry - definition(s).* The advantages of using WordNet, or richer online resources such as thesauri, as the source of a knowledge base, are outlined.

Introduction

Capturing knowledge in a form that is suitable for automated processing has been a long time dream of Artificial Intelligence. For several decades many projects have attempted to capture knowledge by building knowledge bases using complex representations. Semantic Networks were one of the most notable underlying representations used. First order logic also was a primary candidate for such projects. They all, with no exceptions, proved to be very expensive. The high price was mainly due to (1) the un-naturalness of the underlying representation which made it hard for the knowledge engineer(s) to manage an ever increasing repository of hard-to-manage-encoded knowledge and (2) lack of general methods to automatically capture knowledge from existing sources.

In this paper we show a general method to automatically build a large knowledge base at a relativley low price when compared to previous approaches. The method relies on (1) a simple, natural knowledge representation and (2) online dictionaries as a source of common-sense knowledge. As compared to previous work on building knowledge bases our enterprise is limited to common-sense knowledge embedded in the source dictionary. Until later, we use the term online dictionary to refer to any online lexical resource.

The rest of the paper starts with an overview of related work in the next section. Section 3 introduces the Natural Logic Form (NLF), our knowledge representation of choice. The following section presents online lexical resources and then our general method for mapping dictionary entries into knowledge base entries is described. Section 6 applies the method to WordNet and Section 6 describes experiments and results. A *Discussion* section comes next outlining the advantages of applying our method to richer online dictionaries. Conclusions follow.

Related Work

Appropriate computational representations are needed for automated processing of knowledge. In the early days the most successful approaches built systems around *semantic networks* enhanced with efficient reasoning capabilities attached to nodes in the network.

With the advent of TACITUS (Hobbs 1986), there was a change of paradigm in knowledge representation proposed by Hobbs in (Hobbs 1983), in that the representation was based on natural language and efficiencies were removed from the notation for the sake of simplicity.

KL-ONE (Brachman 1985), a language designed for the explicit representation of conceptual information, is based on the idea of *structured inheritance networks.* It was designed for use in the construction of the knowledge base of a reasoning mechanism. LOOM is a knowledge representation system from KL-ONE family. A general characteristic of the KL family of knowledge representation systems is the fact that they embed reasoning mechanism into the net. Those mechanisms lead to a very expensive approach for building knowledge bases.

The Cyc knowledge base (Cyc KB - http://www.cyc.com) is a good example of a large effort to build a common-sense knowledge base. It aims at representing a vast set of facts, rules of thumb, and heuristics for reasoning about the objects and events of everyday life. The knowledge representation consists of the formal language CycL. The knowledge base is organized around the central concept of *microtheory* which is a small set of assertions. Microtheories can be specific to a domain, can focus on a particular interval of time, etc. The concept of a microtheory allows Cyc to contain contradictory assertions. The Cyc KB contains at present time approximately two hundred thousand terms and several dozen hand-entered assertions about each term. New assertions are continually added to the KB by human knowledge engineers.

START is a Question Answering system developed at MIT by Katz (Katz 1997). Given an English sentence containing various relative clauses, appositions, multiple levels of embedding, the START system first breaks it up into smaller units, called kernel sentences (usually containing one verb). Following a separate analysis of each kernel sentence, START rearranges the elements of all parse trees it constructs into a set of embedded representational structures. These structures are made up of a number of fields corresponding to various syntactic parameters in a sentence, but the three most salient parameters, the subject of a sentence, the object, and the relation between them are singled out as playing a special role in indexing. These parameters are explicitly represented in a discrimination network for efficient retrieval.

All previously mentioned systems (except START) attach procedures to nodes in an underlying semantic network. As Hobbs pointed out in (Hobbs 1983) "decoupling the details of implementation, as efficiencies and procedures, from the representation itself would allow to focus on the real problems at hand: in natural language that is interpretation of discourse". This philosophy was used in TACITUS (Hobbs 1986).

TACITUS was a project for interpreting text. Its underlying representation consists of first-order predicate calculus obtained through a large coverage syntactic and semantic translator DIALOGIC, which is an extension of DIAGRAM (Robinson 1982). The main advantage of the representation was simplicity, allowing the researcher to focus on the real problems of text interpretation. DIALOGIC produces a logical form in first-order predicate calculus by encoding everything that can be determined by purely syntactic means, without recourse to context or world knowledge. The advent of TACITUS (Hobbs 1986) produced a change of paradigm in knowledge representation, which started to be visible in the START project, in that the representation was based on natural language and efficiencies were removed from the notation for the sake of simplicity.

Advantages of Natural Language Based Knowledge Representations (KR)

Among the advantages of building NL-based KR systems, as outlined at the first Workshop on Knowledge Representation Systems based on Natural Language (1996 AAAI Fall Symposium), are:

- NL-based systems would be user friendly
- Most human knowledge is encoded and transmitted via natural language and thus NL-based KR are a natural development
- Searching on the Internet has become a necessity and a daily task for most of us. Natural language is heavily used in this task since more than 90% of the web information is textual
- NL-based knowledge processing sytems would provide a uniform symbolic representation for encoding knowledge and processing
- it it is hard to match expressiveness of natural language, particularly in not (well) formalized domains

Recent advances of Natural Language Processing (NLP) in the areas of syntactic parsing, semantics and pragmatics have opened new perspectives for developing expressive KR and building promising NL-based knowledge processing systems.

Our natural language based knowledge representation is the Natural Logic Form. The next section describes in detail the principles standing behind the Natural Logic Form representation. We focus on the syntactic information of predicates, i.e. subject, direct object and others. The implementation approach and details of how to derive logic forms are described in later sections.

The Knowledge Representation

The knowledge representation of choice when building a knowledge base has an impact in all the aspects of the development. We guided our choice by two principles. First, the representation should be close enough to natural language in order to allow us to capture already existing knowledge in textual form. Second, the representation should be close to a logic that is well-understood and has well-defined inference procedures. The Natural Logic Form, a natural language based knowledge representation, seems to best fit our principles.

NLF is first order, syntactically simple, logic and it improves over similar representations used in many language processing applications. Davidson (Davidson 1967) proposed the predicate treatment of verbs and then Hobbs (Hobbs 1983) applied this concept to automated text processing, particularly interpretation of texts. Moldovan and Rus (Moldovan & Rus 2001) proposed a similar representation and used it to Question Answering to search answers that were not explicitly stated in supporting documents.

An example of how an English sentence is mapped onto NLF is illustrated for the following sentence:

The Earth provides the food we eat every day.

Its corresponding natural logic form (NLF) is:

Earth:n_(x1) & provide:v_(e1, x1, x2) & food:n_(x2) & we:n_(x3) & eat:v_(e2, x3, x2; x4) & every:a_(x4) & day:n_(x4)

In NLF a predicate is generated for every noun, verb, adjective or adverb. The name of the predicate is a concatenation of the word's base form and its part-of-speech. In the example given above, the verb *provides* is mapped onto the predicate *provide:v*, where provide is the base form for provides and v stands for verb (the part of speech for provide). Arguments for predicates are of two types: e - for events specified by verbs, respectively x - for entities. Verb tenses and plurals are ignored similar to (Moldovan & Rus 2001). The LF of the entire sentence is the conjunction of individual predicates. Sentences are considered individually, ignoring for the time being discourse issues such as coreference resolution.

The argument's position is also important as it encodes syntactic information: the second argument for a verb is syntactic subject, the third is direct object and the fourth is indirect object. For instance, the second argument of the predicate *provide:v_(e1, x1, x2)* is *x1* (Earth), the subject of the providing event. Arguments after semicolon, ';', are adjuncts (arguments related to time and space which are not mandatory to convey the meaning of a utterance).

NLF has several advantages over similar notations (Montague style semantics, Description Grammars, previous Logic Forms, etc.):

- NLF allows a simple syntax/semantics interface;
- NLF is user friendly - easily readable by new users and thus no special training is needed;
- NLF uses *concept predicates*, resulting in a less ambiguous representation;
- NLF has positional syntactic information that ease other language processing tasks such as textual inference and
- NLF distinguishes between arguments and adjuncts which makes it different from previous representations and more expressive. Argument *x4* of predicate *eat* in the example is an adjunct since it hints the time of the *eating* event and is placed after ';' in the argument list

Once you decided on your KR of choice you can build a knowledge base either manually (see Cyc project) or automatically. The automated option needs a source of knowledge and an automated method to map the source into a knowledge base. In this work, our source(s) are online dictionaries described in the next section. The automated method is presented in Section 5.

Online Lexical Resources

We use the term *online lexical resource* to denote electronic versions of the following three categories: classical dictionaries (Webster's, Oxford's, etc.), thesauri (Roget's) and lexical databases (WordNet). The three categories share the common property of being organized around the elementary structure *word/concept - definition* that constitute an entry in the dictionary. Roget's thesaurus and WordNet have a richer organization of those entries than the simpler lexicographical listing that is used in classical dictionaries.

Dictionaries are a rich source of common sense knowledge since they contain the major concepts (together with their definitions and usage) that people use to understand and reason about the surrounding world. People use them frequently to improve their daily experience of life.

The advent of online versions of traditional dictionaries has opened new posibilities of exploiting their rich content in AI applications. For example, Morris and Hirst (Morris & Hirst 1991) did not implement their algorithm, because there was no machine-readable version or Roget's Thesaurus at that time.

Roget's International Thesaurus, 4th Edition (1977) is composed of 1042 sequentially numbered basic categories. A thesaurus simply groups related words without attempting to explicitly name each relationship.

WordNet is an electronic lexical database developed at Princeton University based on linguistic principles. It is divided into four data files containing data for nouns, verbs, adjectives and adverbs. In WordNet the basic unit is a *synset* - a set of synonymous words which refer to a common semantic concept. Words may be formed in more than one synset: the noun *chocolate* belongs to three different synsets as given in Table 1. The first sense of each word in WordNet is the most frequent sense.

A small textual definition is attached to each synset via a *gloss* relation. An example is also included in the gloss for the large majority of synsets. Table 1 shows a few synsets with their corresponding glosses. WordNet 2.0 contains 115,424 synsets and 152,059 unique words. A list of pointers is attached to each synset and these pointers express relations between synsets. The most important relation is the hypernymy relation. Noun and verb synsets are hierarchically organized based on the hypernymy relation. For example the synset {professor} is a hyponym of the synset {*academician, academic, faculty member*}. In other words, there is a hypernymy relation from the former synset pointing to the latter, and a hyponymy relation in the form of a pointer from the latter synset to the former. Adjectives and adverbs are organized in clusters based on similarity and antonymy relations.

Since dictionary entries' definitions are encoded in plain English it is necessary to transform them into a more computational form that can be processed by other AI tools that require common sense knowledge. The underlying computational representation in this paper is the Natural Logic Form, a first-order, natural language based knowledge representation that contains lexical, categorial, syntactic and semantic information. In this paper we show how dictionary entries can be transformed into world axioms which altogether form a highly inter-connected knowledge base.

Method

Our method takes as input an entry in an online lexical resource and maps it into one or more entries in the knowledge base.

The major steps needed are outlined below.

- extract alternative definitions using explicit markers and exploiting linguistic parallelism cued by conjunctions

Table 1: WordNet 2.0 synsets for word form *chocolate*

Synset	**Gloss**
{cocoa, chocolate, hot chocolate, drinking chocolate}	(a beverage made from cocoa powder and milk and sugar)
{chocolate}	(a food made from roasted ground cacao beans)
{chocolate, coffee, deep brown, umber, burnt umber}	(a medium to dark brown color)

- tokenize the definition, i.e. separate words from punctuation
- expand the definition to full sentence using a technique detailed in the next section
- parse the sentence
- extract the NLF
- propagate arguments from left hand side to right hand side
- generate axiom(s)

In any lexical resource, an entry, be it a word or a concept (set of words with common meaning), has one or more definitions attached to it. Those definitions are in the form of an incomplete sentence. To compensate for the tendency of syntactic parsers to build sentence structures for any given input, our solution is to expand the original definitions of entries.

We give here the algorithm presented to expand entries to full sentences for the case of WordNet definitions, called glosses. Glosses are *expanded* to full sentences using several patterns as described below.

1. For noun glosses, add the first word of the synset, followed by a “be” verb. For example, the definition of {prophet, oracle} becomes *Prophet is an authoritative person who divines the future.*
2. For verb glosses, add *to “verb” is to* to each gloss definition. Example: the definition of {divine} with sense 1 becomes *To divine is to perceive intuitively or through some inexplicable perceptive powe rs.*
3. For adjective glosses, add *it is something* to each definition. Example: The definition of {authoritative, important} becomes *It is something having authority or ascendancy or influence.*
4. For adverb glosses, add *it means* to each definition. For example: the gloss of {intuitively} is extended to *It means in an intuitive manner.*

Although the above patterns are sufficient for most of the cases, a few alternatives are used. For example for noun glosses we use the following alternatives: (1) If noun gloss begins with MD[1] or VBZ add only *noun* as in “Noun performs simple arithmetic functions” or “Noun can be changed to different settings” (2) If the gloss begins with RB, VBN, where, IN add *noun is something* as in “ Noun is something designed to serve a specific function .“ (3) If the gloss begins with plurals NNS, NNPS add *Nouns are* as in “ Nouns are projectiles to be fired from a gun”.

The improvement in parsing accuracy obtained by these expansion techniques can be significant: from 59.77% accuracy on 2,000 raw glosses to 82.25% accuracy on expanded glosses.

Application to WordNet

The previous method can be applied easily to WordNet. One or more axioms are generated from a synset-gloss pair, where the synset is the elementary entry in WordNet and the gloss is the definition of that entry.

The steps of the method as adapted to WordNet are outlined below.

- extract alternative definition for the same entry. In WordNet the end of a definition is marked by “;”.
- tokenize the definition(s)
- expand the definition(s) to full sentences as described in the previous section
- use a part-of-speech tagger and a parser to obtain a structural view of the sentences. We use Eric Brill’s tagger (Brill 1992) and flavors of Michael Collin’s (Collins 1997) parser to obtain syntactic trees for the expanded definition(s)
- generate the natural logic form using structural information embedded in the parse trees. We use an argument-centric approach which given an argument slot for a predicate it predicts the most probable candidate for the slot. We map the argument assignment task into a classification task and use a naive Bayes model to build the classifier. Classifiers are programs that assign a class from a predefined set to an instance or case under consideration based on the values of attributes used to describe this instance. Naive Bayes classifiers use a probabilistic approach, i.e. they try to compute a conditional distribution of classes and then predict the most probable class. To acquire the distributions we use Treebank (Marcus, Santorini, & Marcinkiewicz 1993) ’s, a collection of English text expert-annotated

[1] The meaning of the tags for parts of speech are: MD - modal verb, VBZ - 3rd person singular present, VBN - past participle, RB - adverb, IN - preposition, NNS - common noun plural, NNPS - proper noun plural

with structural information. The Wall Street Journal part contains functional tags in its annotation besides basic phrases such as noun phrase (NP), verb phrase (VP), etc. Those tags were necessary to distinguish words or phrases that belong to one syntactic category and is used for some other function or when it plays a role that is not easily identified without special annotation. In our model we used a set of attributes/features similar to (Blaheta & Johnson 2000) that includes the following: label, parent's label, right sibling label, left sibling label, parent's head pos, head's pos, grandparent's head's pos, parent's head, head. We did not use the alternative head's pos and alternative head (for prepositional phrases that would be the head of the prepositional object) as explicit features but rather we modified the head rules so that the same effect is obtained but in our model is captured in pos and head features, respectively. A simple add-one smoothing method was used. Local dependencies inside a phrase are treated as modifier-modifee relations with the modifee being the head of the phrase.

- propagate arguments from the entry to the definition as described in the rest of this section
- from the logic form and the entry generate entries in the knowledge base. Each such entry can be viewed as an axiom.

A {*synset - definition*} pair can be viewed as an axiom of the underlying concept. For each such pair one or more axioms can be generated: (1) a fact (no left hand side) for synsets with empty definitions (2) a single axiom for the usual gloss (3) many axioms - for definitions that exhibit linguistic parallelism.

There are some specific aspects in the derivation of axioms for each part of speech. Usually a *noun* definition consists of genus and differentia. The template for deriving noun axioms is: *concept(x)* → *genus(x)* & *differentia(x)*. Notice the propagation of arguments from the left hand side to the genus and differentia, without significant syntactic changes.

The *verbs* also exhibit the same structural properties and the derivation is simple for the case of definitions containing only one verb. In the case of definitions consisting of a series of verbs, the derivation of axioms should take care of the syntactic functional changes of the arguments on the right hand side from their counterparts on the left hand side.

Consider veto:v#1 → {*"vote against"*}. The axiom is veto:v#1(e, x, y) → *vote(e1, x, y)* & *against(e1, y)*. One notices the change of *y* from a direct object role for *veto* to a prepositional object role for *vote*. Also event *e* expands in two other events *e1, e2* for the second and third axiom in Table 3.

In the case of *adjectives* which modify nouns, the axioms borrow a virtual head noun as shown here: *American:a#1(x1)* → *of(x1, x2)* & *United_States_of_America(x2)*. Similarly, since *adverbs* modify verbs - their arguments borrow the event of the verb. Adverb fast:r#1 has this corresponding axiom: *fast:r#1(e)* → *quickly(e)*.

Experiments and Results

To validate our procedure we made an experiment on a subset of WordNet 2.0 noun glosses.

We focused on 2,000 entries to be transformed onto axioms. We extracted the corresponding glosses from the WordNet database and for each entry we detected 2,267 definitions. The definitions are then tokenized, expanded into full sentences using the expansion algorithm, part-of-speech tagged and parsed.

The next phase is to generate the logic form. It starts with generating predicates and main arguments for content words that are heads of major phrases. Modifiers of the head words share the argument with the head and are processed next. Then, simple relation predicates are processed such as prepositions, conjunctions etc. More complex predicates, mainly verbs, are processed at last. Their arguments are obtained using a naive Bayes classifier able to select the most probable candidate for a functional role such as logical subject. The classifier is induced using a data set induced from sections 1-21 of Wall Street Journal (WSJ) part of Penn Treebank. The set of attributes/features, presented in the previous section, was automatically extracted from trees together with their classification. In those experiments punctuation was mapped to a unique tag PUNCT and traces [2] were replaced with TRACE.

Two performance measures are reported from (Rus 2002). Each is more useful than the other in some context.

First, we define *predicate level* performance as the number of predicates with correct arguments divided by the total number of predicates. This measure gives a finer-grained look at the derivation process and illustrates the power of a method without considering the application which uses the glosses.

Gloss level performance is defined as the number of full glosses correctly transformed into logic forms divided by the total number of glosses attempted. This new measure catches contextual capabilities of a method in that it gives an idea of how well a method performs at gloss level. It is a more appropriate measure when one tries to see the impact of using full glosses in logic forms to applications such as planning.

On the test set of 2000 glosses we report a performance of 84.9% and 93.2% at gloss and predicate level, respectively.

Except the naive Bayes model induction which require few weeks of development the axiom generation process is fast. For the 2,267 definitions, the axioms were generated in less than half an hour on a regular desktop computer running Linux.

[2] Traces are artificial links introduced in Treebank to accomodate the bracketed representation to remote dependencies.

Table 2: Axioms extracted from the gloss of adjective Romanian:a#1

Romanian(x1) ↔ of(x1, x2) & country(x2) & of(x2, x3) & Romania(x3)
Romanian(x1) ↔ of(x1, x2) & people(x2) & of(x2, x3) & Romania(x3)
Romanian(x1) ↔ of(x1, x2) & language(x2) & of(x2, x3) & Romania(x3)
Romanian(x1) ↔ relate(e1, x1, x2) & country(x2) & of(x2, x3) & Romania(x3)
Romanian(x1) ↔ relate(e1, x1, x2) & people(x2) & of(x2, x3) & Romania(x3)
Romanian(x1) ↔ relate(e1, x1, x2) & language(x2) & of(x2, x3) & Romania(x3)
Romanian(x1) ↔ characteristic_of(x1, x2) & country(x2) & of(x2, x3) & Romania(x3)
Romanian(x1) ↔ characteristic_of(x1, x2) & people(x2) & of(x2, x3) & Romania(x3)
Romanian(x1) ↔ characteristic_of(x1, x2) & language(x2) & of(x2, x3) & Romania(x3)

Table 3: Axioms extracted from the gloss of verb veto:v#1

veto:v#1(e, x, y) ↔ vote(e1, x, y) & against(e1, y)
veto:v#1(e, x, y) ↔ refuse(e1, x, e2) & to(e1, e2) & endorse(e2, x, y)
veto:v#1(e, x, y) ↔ refuse(e1, x, e2) & to(e1, e2) & assent(e2, x, y)

Discussion

We showed in the previous sections how our method of mapping online lexical resources entries onto knowledge base entries can be applied to WordNet. The method can also be applied to any other regular dictionary or thesaurus, such as Roget's. The advantage of applying the method to WordNet, or richer lexical resources than classical dictionaries, is that once you generated the knowledge base entries, you can import from the original resource the rich lexico-semantic relations already available there. For example, for WordNet, once you generated axioms for *abbey*, you have it linked to axioms for its hypernym (superconcept) or hypenyms (subconcepts) and many others. This can be of substantial use in later processing. For a regular dictionary the generated knowledge base is a list of axioms lexicographically ordered. Any use of an axiom may be accompanied by an expensive search.

Conclusions

We presented in this paper a general method to map online lexical resources onto a knowledge base having a natural language based knowledge representation as its underlying coding language. While our experiments were limited to a small part of WordNet they can be easily applied to other resources. The advantages of our method can be summarized as this: (1) it uses a natural language based knowledge representation that makes the management much easier (2) it allows the aqcuisition of knowledge from textual form where most of the knowledge is already available.

References

Blaheta, D., and Johnson, M. 2000. Assigning function tags to parsed text. In *Proceedings of the 1st Annual Meeting of the North American Chapter of the Association for Computational Linguistics, Seattle, May 2000, pp. 234-240.*

Brachman, R. 1985. An overview of the KL-ONE knowledge representation system. *Cognitive Science* (9):171–216.

Brill, E. 1992. A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, 152–155.

Collins, M. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the ACL.*

Davidson, D. 1967. The logical form of action sentences. In Rescher, N., ed., *The Logic of Decision and Action.* University of Pittsburgh Press. 81–95.

Hobbs, J. 1983. Ontological promiscuity. In *Proceedings 23rd Annual Meeting of the ACL*, 57–63.

Hobbs, J. R. 1986. Overview of the TACITUS project. *Computational Linguistics* 12(3).

Katz, B. 1997. From sentence processing to information access on the world wide web. In AAAI., ed., *Proceedings of AAAI Spring Symposium on Natural Language Processing for the World Wide Web.*

Marcus, M.; Santorini, B.; and Marcinkiewicz. 1993. Building a large annotated coprus of english: the penn treebank. *Computational Linguistic* 19(2):313–330.

Moldovan, D. I., and Rus, V. 2001. Logic Form transformation of wordNet and its Applicability to Question Answering. In *Proceedings of ACL 2001.* Toulouse, France: Association for Computational Linguistics.

Morris, J., and Hirst, G. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics* 17(1):21–43.

Robinson, J. 1982. DIAGRAM: A grammar for dialogues. *Communications of the ACM* 25(1):27–47.

Rus, V. 2002. High precision logic form transformation. In *International Journal for Tools with Artificial Intelligence.* IEEE Computer Society.