Proceedings of the 5th WSEAS International Conference on Telecommunications and Informatics, Istanbul, Turkey, May 27-29, 2006 (pp343-347)

Localist Approach to Natural Language Processing

MLADEN STANOJEVIĆ, SANJA VRANEŠ
The Mihailo Pupin Institute
Volgina 15, 11060 Belgrade
SERBIA AND MONTENEGRO
www.institutepupin.com

Abstract: - Determining the meaning of a sentence in a natural language requires an efficient search mechanism to find the corresponding point in a very large space of sentence meanings. Examination of large search spaces must rely on constraints to guide the search process and provide satisfactory performance. However, existing knowledge representation techniques used in AI, both classical and connectionist, do not satisfy all the requirements needed to represent adequately the semantics, whereby neither of these techniques meet satisfactorily the most important requirement – proper context representation. We propose a knowledge representation technique named Hierarchical Semantic Form that can be used, together with the Space Of Universal Links (SOUL) algorithm, to represent the semantics adequately. To check the viability of the proposed solution we have implemented a prototype Semantic Web service that provides information about flight timetables, defined in a natural language within an ordinary HTML page, using natural language queries.

Key-Words: - Natural Language Processing, Localist Approach, Connectionist Model, Knowledge Representation

1 Introduction

Almost from the beginning of the Artificial Intelligence era, researchers were divided into two strongly opposed groups, supporters of classical approach and supporters of connectionism (artificial neural networks) [1]. Classicists were influenced by the ideas of Universal Turing Machine and von Neumann computer architecture, while connectionists were inspired by the organization of neural system and brain and the principles of their functioning. The main results of the classical approach are in the simulation of higher cognitive functions that require symbolic processing like reasoning, problem solving and simulating human behavior, whereas connectionist approach is applied mainly in machine learning, pattern recognition, computer vision and syntactic parsing.

The main objection to the classical approach is that it cannot be used for the large-scale modeling of real world, because such models require representation and search of large spaces, which cannot be achieved by sequential processing. On the other hand, classicists argue that the connectionist model cannot represent adequately a combinatorial syntactic and semantic structure [2].

Classicists employ different knowledge representation techniques to simulate the intelligent behavior of AI applications. They use semantic nets, frames, object-attribute-value triplets and logical formalism to describe the domain of application [3], and rules or predicate calculus to define inference procedures. However, these knowledge representation techniques are specialized to represent one type of knowledge and cannot be used to represent efficiently the meaning of sentences.

Although some attempts have been made to introduce semantic grammars, syntactic grammars are mainly used today in both classical and connectionist approach. This was probably caused by a deceptive possibility that a relatively simple, general syntactic grammar capable of parsing any sentence in a natural language is within our reach. However, even if such grammar is proposed one day, it will not contribute a lot to Natural Language Processing, because sentences that share the same syntactic structure may have completely different meanings.

To decrease the complexity of the problem of Natural Language Understanding, some solutions propose representation and use of common sense knowledge in the form of frequently used phrases [4], while some others are using genetic programming to provide classification in Web mining applications [5].

Based on the connectionist approach with localist representation, we have developed a Hierarchical Semantic Form (HSF) with the supporting Space Of Universal Links (SOUL) algorithm that could be used to represent more types of knowledge than traditional, specialized techniques as well as support tasks such as Natural Language Processing, learning or reasoning. HSF resembles natural, hierarchically organized neural networks.

To verify the ideas applied in HSF and SOUL, we have implemented a prototype Semantic Web service [6] that gives information about flight timetables. A user

Proceedings of the 5th WSEAS International Conference on Telecommunications and Informatics, Istanbul, Turkey, May 27-29, 2006 (pp343-347)

makes a natural language query about a flight, and Web service then finds all flights that satisfy the query.

2 Natural Language Processing

The problem of understanding sentences written in a natural language is in its essence a search problem. The problem is how to map a sentence, which is, for example, 100 characters long, to a point representing the meaning of that sentence in the space of all meaningful sentences.

A brute force approach would be to attach meaning to all meaningful sentences and when a sentence is fed to system to try to match the sequence of characters representing the given sentence to sequences of characters representing all meaningful sentences. However, the brute force approach is not practical, because the search space for sequences representing sentences is extremely large (e.g., for a 100 character long sentence consisting of letters and blank characters, the search space contains $100^{27} = 10^{54}$ points). However, this space is very sparsely populated (i.e. it includes sequences of blank characters and many other meaningless sequences), hence a good constraint mechanism (proper context representation) could enhance search a lot.

As is well known in the engineering practice, the complexity of a problem can be decreased if it is decomposed into less complex subproblems. In AI this reduction of complexity is achieved using two principal relations, *part-of* and *is-a*, used to create hierarchies of constituents and classes. The whole universe can be observed as a part-whole hierarchy where complex objects are comprised of other objects (parts). On the other hand, human ability to abstract common characteristics of similar objects and to create hierarchies of classes is indispensable for our intelligent behavior.

The same approach can be used in Natural Language Processing. The part-whole hierarchy can easily be applied to sequences of characters representing sentences: syllables are sequences of letters, words are sequences of syllables, groups of words are sequences of words and sentences are sequences of groups of words. The search space of sentence meanings contains subspaces of groups of words, words and syllables. Note that search spaces at each level of hierarchy are relatively small (a syllable contains a few letters, a word contains a few syllables, a group of words contains a few words, etc.).

The other type of hierarchies, class hierarchy, has been applied in Natural Language Processing mainly on a syntactic level. Although syntactic generalization can be successfully applied to determine the syntax of a sentence, it cannot be used to find its meaning. The main problem lies in the fact that the sentences with the same syntax may have completely different meanings. For example, sentences “John loves Mary” and “Fido chases Felix”, where Fido is a dog and Felix is a cat, share the same syntax, although their meanings are quite different.

Another important issue that has to be solved in Natural Language Processing is search efficiency. At the level of words, a simple hashing mechanism can be used to match words. This can easily be done for a relatively small search space of words that contains a few hundred thousands of words. However, difficulties arise progressively when the hashing mechanism has to be applied to match groups of words or sentences, because the corresponding search spaces are substantially larger.

An alternative approach would be to record the context for each category, which is defined by the preceding category and all successor categories. If we for all categories provide the unique representation, which is an important requirement related to representational and search efficiency, we can implement an efficient constraint mechanism that will allow us to examine meaningful sequences only. When we match a category (be it a letter, syllable, word, group of words or semantic category), we can trace its immediate successor to find the next match. This is much like finite automata or Augmented Transition Networks.

In brief, a knowledge representation technique suitable for Natural Language Processing should provide the following:

- representation of part-whole hierarchies
- representation of class hierarchies
- unique representation of categories at different levels of abstraction (letters, syllables, words, groups of words, semantic categories, sentences)
- context representation

Knowledge representation techniques used in classicist approach (frames, semantic nets, object-attribute-value triplets, rules, logical formalism) can be used to represent part-whole and class hierarchies, some of them provide unique representation of categories, but neither of them supports proper context representation. Without context representation, they are not capable of representing a sequence in different contexts, for instance complex semantic categories, which represent sequences of words and other semantic categories. Frames, semantic nets and object-attribute-value triplets can represent simple semantic categories in the form, “Apple is a fruit”, but they cannot represent properly complex semantic categories defined using the following grammar rules:

```
<departure-phrase> ::= <depart> <part-of-day>
                                   <from-phrase>
```

```
<arrival-phrase> ::= <arrive> <part-of-day>
                                 <to-phrase>
```

On the other hand, rules and logic formalism are capable of representing these semantic categories, but at the price of representational and search inefficiency, because they do not allow unique representation (e.g. for <part-of-day>).

The connectionist approach (artificial neural networks) offers representation of part-whole and class hierarchies and unique representation of categories. However, in the connectionist model, more general classes are lower in the unit hierarchy. They cannot represent semantic categories, nor provide semantic generalization. Furthermore, units at the lower level of hierarchy are linked only with the units which are one level higher. Units at the same level of hierarchy cannot be linked with each other. This causes difficulties in the representation of sequences and context representation. Some rather artificial solutions of this problem have been reported, like introducing separate sequence numbering for each level of hierarchy, role-filler binding [7] or syntactic parsing [8].

3 Hierarchical Semantic Form

Researchers in AI are divided into classicists and connectionists, and the connectionists themselves are further divided into the supporters of localist representation and the supporters of distributed representation. In localist representation, units describe items uniquely (subsymbols and symbols), while in distributed representation, items are represented using vectors of units (codevectors). Localist representation is intuitive, because each unit represents a subsymbol or symbol allowing means for easy symbolic processing. However, the supporters of distributive approach object that localist representation is not suitable for complex problems, because the number of units grows exponentially with the number of hierarchy levels.

We think that localist representation is better suited for knowledge representation in Natural Language Processing than distributed representation, because it provides unique representation of subsymbols and symbols. The objection of the exponential growth of units doesn't stand in case of Natural Language Processing, because the complexity of the space of meanings can be kept under control by using semantic generalization. However, pure localist representation is not capable of supporting semantic generalization or proper context representation, hence we propose Hierarchical Semantic Form (HSF).

Patterns appearing in the flight timetable example (and in Natural Language Processing in general) are in their essence sequences. Patterns at the lowest level of hierarchy are characters, syllables are sequences of characters, words are sequences of syllables, groups of words are sequences of words, semantic categories are sequences of words and other semantic categories, while queries and flight definitions are sequences of semantic categories. Except at the lowest level of hierarchy, complex patterns represent sequences of simpler patterns.

HSF is comprised of two types of units, *groups* and *links*, which enable unique representation of patterns and hierarchical organization of sequences. The group unit (Fig. 1.a) is similar to units in localist representation, which designates characters, groups of characters, words, semantic categories, queries and flight definitions. This data abstraction is used to represent sequences at different levels of hierarchy (a group points to the first link of a sequence). One group may appear in different contexts, so it can have many associated links (for each context – one link). This way a unique representation of patterns is provided.

The link unit (Fig. 1.b) enables the creation of sequences at different hierarchy levels. This type of units is not present in pure localist representation and, unlike ordinary units, they are linked with each other at the same hierarchical level, thus providing intuitive and explicit sequence representation. The main role of links is to represent patterns (groups) in different contexts. For each new context where a pattern appears, we need a new link. A link points to the group it represents within the sequence, but also to a preceding link and all successive links (defining the context of the pattern). If a link is the last in the sequence of links, instead to successive links it points to a group that represents this sequence.

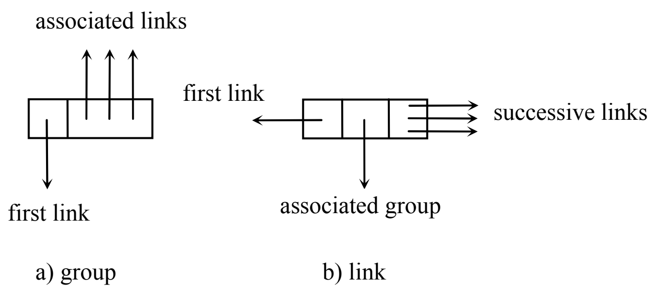

Fig. 1. HSF data types

In the beginning HSF consists of only predefined groups representing characters (patterns at the lowest level). If we then feed words to SOUL algorithm, HSF will grow by making patterns representing these words.

Notice that HSF represents a hierarchical equivalent of plain text (flat form). A plain text can be uniquely translated into a hierarchical form, and vice versa with no information losses. The difference between plain text form and hierarchical form is that in the hierarchical form all patterns and relationships between them are explicitly represented.

We use SOUL Commander, a kind of Natural Language Processing shell, to define semantic categories of various complexities. Semantic categories are represented in HSF using <is-a> keyword (Fig. 2).

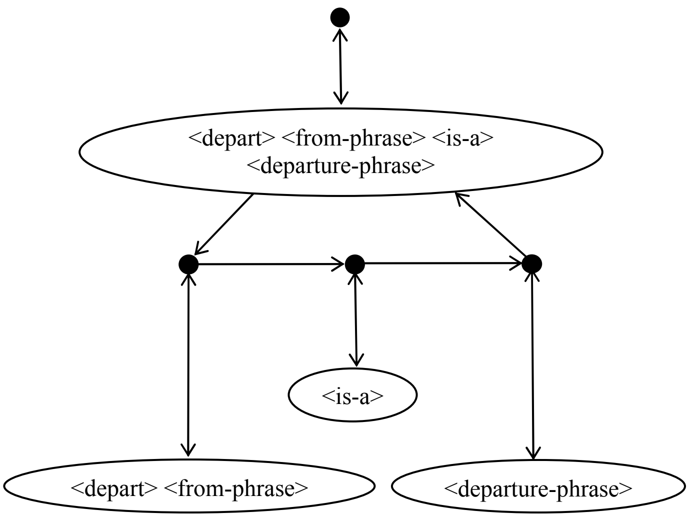

Fig. 2. Semantic category in HSF

We can easily extend the matching capabilities of <departure-phrase> semantic category by defining new instances of <town> semantic category (e.g., London, Berlin, New York, Tokyo), <depart> (e.g., departs, depart, departure), or by defining new forms of <departure-phrase> semantic category (e.g. <departure> <part-of-day> <from-phrase>).

4 SOUL algorithm

Space Of Universal Links (SOUL) algorithm is capable of learning new patterns, new semantic categories and their instances. When we feed plain text to it, SOUL algorithm performs partial matching using the existing patterns and semantic categories defined in HSF, discovers old patterns in new text, creates new patterns (if there are any), performs matching of existing semantic categories and finally creates a HSF representation of new text consisting of old and new patterns and semantic categories. Unlike other connectionist solutions, which can learn a structure when structures are fed to them, HSF and SOUL algorithm support unsupervised learning of structures from plain text.

Unique representation of patterns and semantic categories gives rise to the learning capability of SOUL algorithm. SOUL acts as a bottom-up parser, which performs partial matching able to locate the existing patterns, and discover new patterns if there is a sequence of existing patterns that matches a part of new text.

There are three possible cases when new patterns can be discovered: at the beginning of a sequence (Fig. 3.a), in the middle of a sequence (Fig. 3.b), and at the end of a sequence (Fig. 3.c). When SOUL algorithm discovers a new pattern, a new group, that will uniquely represent this pattern, is created as well as two new links representing this new pattern within two separate contexts. This way SOUL supports unique pattern representation in all contexts.

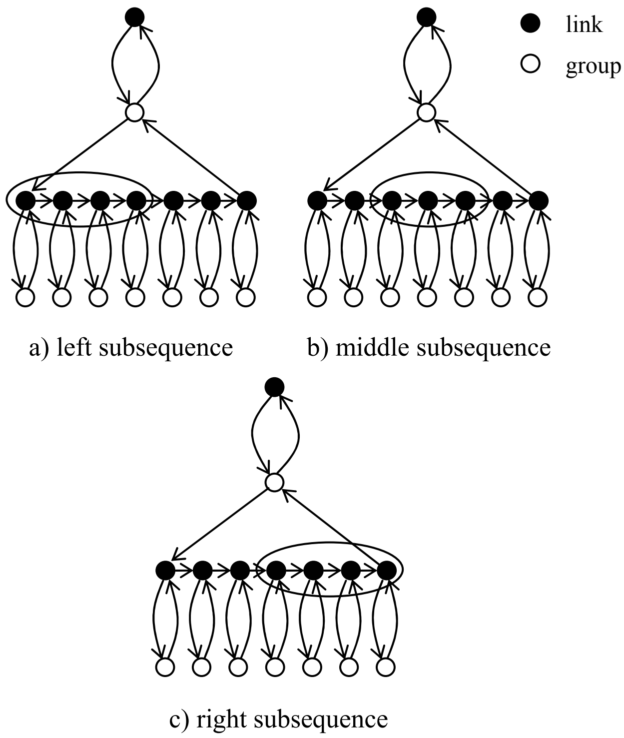

Fig. 3. Pattern discovery

5 Semantic Web service example

Web researchers have introduced Semantic Web [7] in an attempt to overcome the inability of computer programs to understand the content of ordinary Web pages. To make computer programs understand Web pages, they must be annotated using one of ontology or schema languages (e.g., XOL, SHOE, OML, RDFS, DAML+OIL, OWL), which requires translation of existing Web pages using the chosen knowledge representation formalism. Furthermore, the implementation of Semantic Web services requires use of different knowledge representation techniques, which gives rise to some integration problems.

We were inspired by the ATIS (Air Travel Information System) project launched by ARPA in 1988 to develop a prototype Flight Information Service (FIS), a Semantic Web service. The idea in ATIS project was to develop an interactive system for querying the ATIS database and essentially going through all the steps it would take to book a real flight. We have transferred the problem to Semantic Web domain and simplified it so that FIS provides only information about flight timetables.

We implemented a repository of flight timetables in natural language for major European airlines in an ordinary HTML page. Using SOUL Commander we have defined semantic categories used to understand flight definitions as well as queries about flights.

FIS is able to process complex natural language queries about flights, but is also able to communicate with a user in a simple dialog form where it tries to collect the basic information about the needed flight (departure and arrival city and day/date of flight).

FIS is storing the dialog context, which enables the processing of partial queries like:

"Give me the first two flights only!"
"Departing at 7 o'clock in the morning"
"Departing on next Sunday"
"What are the AlItalia flights?"

Although FIS ignores words it doesn't understand, and performs partial parsing, which provides a great flexibility in understanding user queries, its understanding capability is limited by the used semantic categories. However, new semantic categories can easily be added to provide better understanding.

5 Conclusions

Natural Language Processing is in its essence a search problem, where a point representing the meaning of sentence must be found in a very large space of all possible meanings. In decreasing the problem complexity, part-whole and class hierarchies can be of great help, while unique representation of patterns (letters, syllables, words, groups of words, semantic categories and sentences) at different levels of hierarchy, and proper context representation, provide efficient search.

By improving the localist representation of connectionist model, we have developed Hierarchical Semantic Form (HSF) with the corresponding Space Of Universal Link (SOUL) algorithm that satisfies all requirements needed for Natural Language Processing.

To check the basic ideas underlying HSF and SOUL algorithm, we have implemented a prototype Semantic Web service, Flight Information System (FIS), which provides information about flight timetables using natural language queries. HSF combined with SOUL algorithm solves the main integration problem in the implementation of Semantic Web services (translation of existing Web pages), because they support Natural Language Processing and do not require use of annotated pages. In FIS example, we defined the repository of flight timetables as an ordinary HTML file using natural language. FIS is scalable and portable, so its understanding capabilities can easily be upgraded using new semantic categories, while definitions of FIS semantic categories can easily be reused, for instance, in a system providing information about bus or train timetables.

References:

[1] D. Medler, "A Brief History of Connectionism", *Neural Computing Surveys*, Vol. 1, No. 2, 1998, pp. 18-72.

[2] J. Fodor and Z. Pylyshyn, "Connectionism and Cognitive Architecture: A Critical Analysis", *Cognition*, No. 28, 1988, pp. 3-71.

[3] S. Vraneš, M. Stanojević, "Prolog/Rex - A Way to Extend Prolog for Better Knowledge Representation". IEEE Transactions on Knowledge and Data Engineering, Vol. 6, No. 1, 1994, pp. 22-37.

[4] R. Spiegel, "A Machine Learning Approach towards Improving Internet Search with a Question-Answering System", *4th WSEAS Int. Conf. on COMPUTATIONAL INTELLIGENCE, MAN-MACHINE SYSTEMS and CYBERNETICS* Miami, Florida, USA, November 2005, pp. 270-275.

[5] J. Aguilar and J. Altamiranda, "A Genetic Programming Based Algorithm for Web Mining", *Proceedings of the 6th WSEAS Int. Conf. on EVOLUTIONARY COMPUTING*, Lisbon, Portugal, June 2005, pp. 98-105.

[6] M. Stanojević, S. Vraneš, „Semantic Web Services with Soul", *Lecture Notes in Computer Science*, Vol. 3704, 2005, pp. 338-346.

[7] P. Lane and J. Henderson, "Incremental Syntactic Parsing of Natural Language Corpora with Simple Synchrony Networks", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 13, No. 2, 2001, pp. 219-231.

[8] D. Rachkovskij, "Representation and Processing of Structures with Binary Sparse Distributed Codes", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 13, No. 2, 2001, pp. 261-276.

[9] D. Fensel, J.A. Hendler, H. Lieberman and W. Wahlster, *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, MIT Press, 2003.