INTERPRETATION OF USER UTTERANCES

Miloslav Konopík and Roman Mouček

University of West Bohemia,
Department of Computer Science and Engineering
Univerzitní 22, CZ -306 14 Plzeň, Czech Republic

Abstract: In this article we present a novel approach to the interpretation of user utterances in order to understand the content of an utterance. We are focused on conversational spoken language. The spoken language shares many features with the written language but it also differs in a number of aspects. We specialize in restricted domains. The spoken languages and restricted domains require considerable modification of existing algorithms for the general written language. In this paper we show how to extract knowledge from an utterance transcription.

Keywords: interpretation, natural language understanding, morphological tagging, syntactic analysis, semantic analysis

1. INTRODUCTION

In this article we discuss the interpretation problems. Every system which accepts new knowledge from its environment has to deal with interpretation. Interpretation is the process of incorporating new information into old knowledge. If we work with spoken input we sometimes call this process the natural language understanding problem. We are trying to develop a system capable of understanding human speech and working with knowledge taken from an utterance.

The system which is described in this paper is a part of CIDS (City Information Dialogue System). All examples presented in this article are taken from CIDS corpus and translated from Czech.

In this paper we will assume that the utterance is already transcribed from the spoken form to the text representation. This transcription is a task for the ASR[1]. The result of the ASR differs significantly from the written text (e.g. from a paper, internet, etc). The structure of a transcription is completely different from a text. It contains disruptive elements, hesitations (um …), errors, repairs, non-syntactic constructs, etc.

[1] ASR = Automatic Speech Recognition

1.1. Utterance processing

We gradually deal with more and more complicated structures during the utterance processing. At first we work with individual words. We need the knowledge about morphology, which captures the information about the shape and behaviour of words in a context. We call the process determining the morphological categories the morphological tagging. It is also useful to know the lemma[2] of each word. At last but not at least we need to know the meaning of a word. The process of distinguishing of word meanings is called the word sense disambiguation. If we only label the meaning of a word with a limited set of tags we call it semantic tagging.

The following step in the utterance processing is called syntactic and semantic parsing. In this step we work with the whole sentence and we create a tree representation of the utterance.

The last step in the utterance understanding is the interpretation. During this phase we need to mimic the human thinking to derive new knowledge and to maintain a valid knowledge model. We work with a parsed sentence and we use the knowledge about the world and the dialog history.

2. TAGGING

2.1. Morphological tagging

As we mentioned in chapter 1.1 it is useful to know the shape and behavior of a word. These features of a word are encoded in morphological categories[3]. During the morphological tagging we assign a morphological tag to each word (see the example in Fig. 1).

I want to travel to Prague at six o'clock

PRP VBP TO VB TO NNP IN CD NN

Fig. 1. Example of morphological tagging

The list of all possible tags is stored in the tagset. In this example we used the Penn Tree bank tagset (Marcus, *et al*., 1993). An example of this tagset is presented in the Table 1. There are listed only those tags we used in Fig. 1. For complete list see (Marcus, *et al*., 1993).

Table 1. Penn Tree bank tagset example

Tag	Explanation
PRP	personal pronoun
VBP	verb, sing. present, non-3d
TO	To
VB	verb, base form
NNP	proper noun, singular
IN	Preposition /subordinating conjunction
CD	cardinal number
NN	noun, singular or mass

[2] Lemma = the basic from of a word

[3] We have made recently an experiment with automatic word clustering. This clustering was based on the similarities in behavior of words (we counted the mutual information of adjacent words, the candidates to merge were the words which caused the minimal loss of total mutual information). It showed up that the resulting classes agree more or less to the parts-of-speech.

2.2. Solution for morphological tagging

We can use several methods based on different principles to accomplish the tagging task. The most commonly used methods for tagging are: HMM (Hidden Markov Models) (Jurafsky and Martin, 2000), Rule based – Brill's tagger (Brill, 1995), Maximum Entropy taggers (Ratnaparkhi, 1996), Feature-based taggers, etc.

We use the HMM tagger because of its performance and simplicity. The noisy channel model (Fig. 2) can be used for modeling the HMM tagging process. This model says that we see the output (the word forms) and we want to find out the input (tags) in presence of the noise, which was added by the channel.

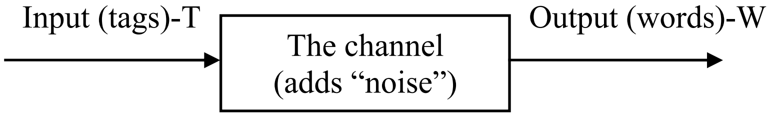


Fig. 2: The noisy channel model

The equation 1 shows that HMM tagging is the application of the Bayesian rule:

$$\begin{aligned} &\mathrm{p}(T \mid W) = \mathrm{p}(W \mid T)\mathrm{p}(T) / \mathrm{p}(W) \\ &t_{best} = \arg\max_T \mathrm{p}(W \mid T)\mathrm{p}(T) \end{aligned} \tag{1}$$

A direct computation (finding the t_{best}) of these equations is unfeasible because it would be necessary to try every possible sequence of tags T (it has the exponential complexity). Instead of a direct computation we use the Viterbi algorithm, which is based on an application of dynamic programming. This algorithm is described in (Manning and Schütze, 2001) in detail.

2.3. Semantic tagging

In addition to morphological tagging we use semantic tagging to label the meaning of a word. Semantic tagging is not common in the field of natural language processing but we found out that it significantly increases the successfulness of the parsing.

The process of the semantic tagging is very similar to the morphological tagging with one exception. The tagset for morphological tagging is task independent but the tagset for semantic tagging has to be task dedicated.

The Fig. 3 shows an example of semantic tagging. The related tagset is in the Table 2.

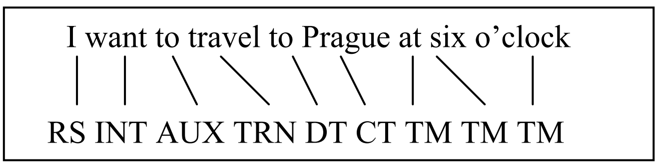


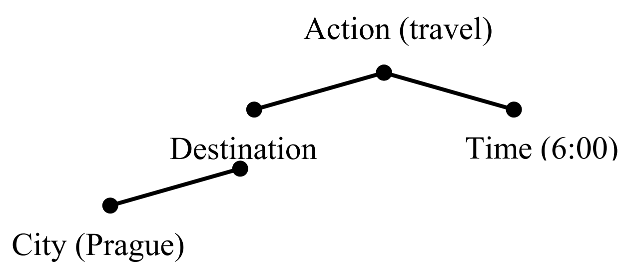
Fig. 3. Example of semantic tagging

Table 2. Semantic tagset example

Tag	Explanation
RS	Reference to the Speaker
INT	Intention
AUX	Auxiliary
TRN	Transport
DT	Direction To
CT	City
TM	Time

3. SYNTACTIC AND SEMANTIC PARSING

Now, let's move from words to sentences and build a more complex structure. There is also difference between written input and spoken input. In written input we usually use the syntactic analysis to create a syntactic parse tree[4]. But the syntax itself cannot provide a reliable source of information in a spoken input. It is because of irregularities and disfluencies in a spoken input (see Eckert and Niemann (1994)). Therefore it seems to be useful to join the syntactic and semantic analysis together. There is another approach to deal with the spoken language and it is the partial parsing[5]. We use the joined syntactic and semantic parsing because we do not believe in syntax in a spoken input. The result of a semantic analysis has usually a tree form (see Fig. 4).



"to travel to Prague at six o'clock"

Fig. 4. Example of semantic analysis result

The best formalism for capturing the structure of a sentence seems to be the context-free grammar. Although this formalism is not usually used in its base form, we rather define the base form in this article in an effort not to confuse the reader.

Definition (Context-free grammar): A context-free grammar is a 4-tuple G = (N, Σ, R, S):

1. a set of non-terminal symbols (or "variables") N
2. a set of terminal symbols Σ (disjoint from N)
3. s set of productions R, each of the from $A \rightarrow \alpha$, where A is a non-terminal and α is a string of symbols from the infinite set of strings $(\Sigma \cup N)^*$
4. a designated start symbol S

A language is defined via the concept of derivation. One string derives another one if it can be rewritten as the second one via some series of rule $r_i \in R$ applications.

Definition (derivation): Let $\alpha_1, \alpha_2, \ldots, \alpha_m$ be strings in $(\Sigma \cup N)^*, m \geq 1$,such that

$$\alpha_1 \Rightarrow \alpha_2, \alpha_2 \Rightarrow \alpha_3, \ldots, \alpha_{m-1} \Rightarrow \alpha_m$$

We say that α_1 derives α_m, or $\alpha_1 \overset{*}{\Rightarrow} \alpha_m$.

The context-free grammar is usually extended via features (see chapter 4 in (Allen, 1995)) to capture natural language properties. We use the stochastic extension. In this extension every rule has a conditioned probability of its application. We can redefine the context-free grammar definition to capture the probability:

---

[4] Nodes in a syntactic parse tree represent phrases (e.g. verb phrase, noun phrase) and relations has a syntactic function (e.g. a sentence composes of the noun phrase and the verb phrase …).

[5] If the utterance can not be parsed as a whole, it is divided into several parts. These parts are parsed separately. It is the goal for the interpretation module to join these fragments.

Definition (Stochastic context-free grammar): The same as the definition of context-free grammar, instead of:

3. a set of productions R, each of the from $A \rightarrow \alpha$ $[p]$, where A is a non-terminal and α is a string of symbols from the infinite set of strings $(\Sigma \cup N)^*$, p is the probability of a rule application $R(A \rightarrow \alpha \mid A, c_1, c_{2, \ldots,} c_m)$, where c_i are the conditions.

The c_i are the conditions of rule application – the morphological tag, the semantic tag, the words itself, etc.

We use the probabilistic bottom-up chart parser to parse a sentence. For more details about stochastic context free parsing please consult (Manning and Schütze, 2001), chapter 11.

4. INTERPRETATION

At this point we have the structured information about the utterance. And it is time to gain more knowledge which is not explicitly mentioned in the utterance. Interpretation should use similar cognitive actions as a man would to gather this knowledge. It stands that the better the semantic interpretation is the more clever the system we get. "More clever" in this context means that the system is more similar to a human.

The main assumption made about semantic interpretation is that it is a compositional process[6]. This means that the meaning of a constituent is derived solely from its sub-constituents (Allen, 1995). We use this theory because of its attractive properties. The main advantage is that we can build interpretation of an utterance incrementally from its sub-phrases. Then the inference rules are much simpler and can deal with problems at separated levels. One of the formalisms of the theory of compositionality is based on the lambda calculus (chapter 9 in (Allen, 1995)).

The interpretation module is usually an expert system. It uses reasoning techniques to work with knowledge. During reasoning the system has to use the default knowledge (to add obvious but not mentioned information, e.g. that Prague is in Czech Republic). According to the theory of compositionality the knowledge from sub-constituents is linked together by the superior constituent. The relations between particular knowledge can produce new knowledge (it is called the derived knowledge). The Fig. 5 shows the result of the interpretation for our practice utterance. The type of knowledge is marked according to the legend.

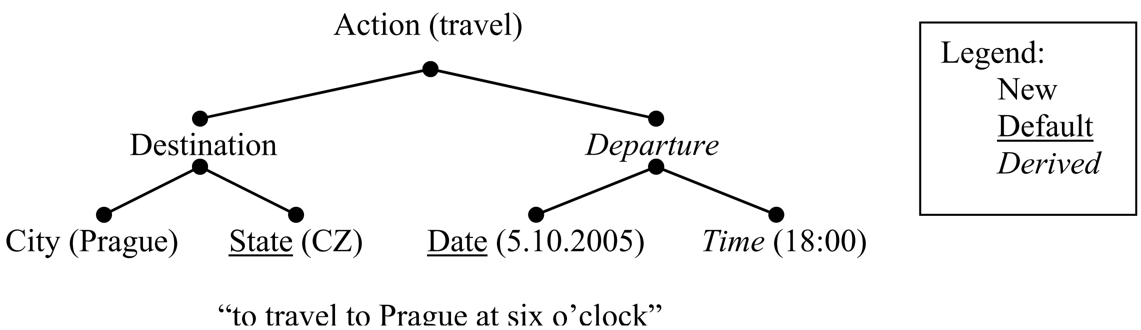


Fig. 5. Interpretation example

[6] Actually, this assumption is already made about context-free grammars (a sub-phrase does not depend on its context).

We developed a declarative-procedural interpretation system based mainly on principles of the SIL[7] language (Peckham, 1993). We store the procedural inference rules in a declarative structure. The knowledge is represented by a concept dependency structure. Each concept represented as a Java class is treated as an autonomous agent. Every agent (= concept) reacts to a stimulant by activating its rules. The stimulant is a notification about a change of any sub-constituent, which the agent is dependent on. The position in the knowledge structure is stored declaratively as well as the concept dependency. On the other hand, the rules are written in Java code, but they can use many services of declarative structure (e.g. notify somebody about a change, overlay knowledge, recall the history and others).

Our system incorporates the automatic anchoring algorithm. The system accepts predictions from the dialog manager and tries to anchor the knowledge in the structure. See (Konopík and Mouček, 2005) for more details about our interpretation system.

5. CONCLUSION

According to the (Mouček, 2004) article it is virtually impossible to achieve a successful interpretation in an unrestricted domain. Thus our goal is to develop a set of tools capable of interpretation in a restricted domain. Our tools are based on machine learning so it is not necessary to change or develop new algorithms when switching to another domain. It should be sufficient to provide the system with different training data.

REFERENCES

Allen, J. (1995). *Natural Language Understanding*. Benjamin/Cummings Publ. Comp. Inc., Redwood City, California.

Brill, E. (1995). *Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging*. Computational Linguistics, The Johns Hopkins University.

Eckert, W. and Niemann, H. (1994). Semantic analysis in a Robust Dialog System, *Proc. Int. Conf. on Spoken Language Processing,* pages 107 – 110, Yokohama.

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing*. Prentice-Hall Inc., New Jersey.

Konopík, M. and Mouček, R. (2005). An Alternative Way of Semantic Interpretation. In: *Proc. Int. Conf. on Text, Speech and Dialogue* (Matoušek, V., Mautner, P., Pavelka T.), pages 348 – 355, Springer, Karlovy Vary.

Manning, Ch. D. and Schütze, H. (2001). Foundations of Statistical Natural Language Processing. The MIT Press, London.

Marcus, M. P., Santorini, B., Marcinkiewicz, M. A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank, *Computational Linguistics*, pages 313 – 330.

Mouček, R (2004). Semantics in Dialogue Systems. *Doctoral Thesis.* Pilsen.

Peckham, J. (1993). A New Generation of Spoken Dialogue Systems: Results and Lessons from the SUNDIAL Project, *In. Proceedings of the EUROSPEECH ’93,* pages 33-40, Berlin.

Ratnaparkhi, A. (1996). *A Maximum Entropy Model for Part-Of-Speech Tagging.* Proceedings of the First Empirical Methods in Natural Language Processing Conference. Philadelphia, Pa.

[7] SIL = Semantic interface language