

# Conceptual Language Models for Dialog systems

**Renato De Mori**

LIA CNRS BP 1228

84911 Avignon Cedex 9 - France

renato.demori, @lia.univ-avignon.fr

**Frederic Béchet**

LIA CNRS BP 1228

84911 Avignon Cedex 9 - France

frederic.bechet, @lia.univ-avignon.fr

## 1 Introduction

The purpose of computer speech understanding is to find conceptual representations from signs coded into the speech signal.

Contrary to speech interpretation by humans in which the same discourse may be interpreted differently by different subjects, for practical applications of computer understanding the result of interpretation should be unique for a given signal. Usually it is represented by an object which is an instance of class corresponding to a semantic structure which can be fairly complex even if it is built with instances of conceptual constituents belonging to a small set of major ontological categories.

The mapping process that leads to a semantic interpretation can be derived manually because human interpretation of sentences can be completely explained with a logical formalism or it can be inferred by machine learning algorithms in order to ensure a large coverage of possible sentence patterns. Theories and practical implementations of these approaches are proposed in [1],[2].

Limitations of coverage in the manual approach and in precision of machine learning can be reduced by making manually a detailed analysis of a limited number of examples and generalizing each analysis with automatic methods. In particular, a well structured lexicon can be very useful, in which the meaning of words is represented together with suggestions of possible syntactic and conceptual structures.

Word associations found with networks of word relations [3] can also be useful for suggesting compositions of semantic constituents into conceptual structures. Thus, given an observed example, other examples can be manually derived and generalized automatically.

Computer understanding of a spoken sentences is problem solving activity whose central engine is a search process involving various types of models.

Searching for concepts can be combined with searching for words. This suggests that statistical language models

(LMs) could be adapted based on expectations of concepts predicted by a system belief. With this perspective, it is important to notice that, while the observation of only certain words may be sufficient for hypothesizing a conceptual structure, complete details of word phrases expressing a conceptual structure have to be known in order to adapt a generic LM to the expectation of such a structure.

This paper introduces a search method and a learning paradigm based on the just introduced considerations.

The search engine built with this method finds the best common path between the system knowledge represented by the composition of Stochastic Finite State Transducers (SFST) and a Stochastic Finite State Automaton (SFSA) representing the lattice of word hypotheses generated by an Automatic Speech Recognition System (ASR).

## 2 Hypothesis evaluation and search

Let a dialogue system have a belief which generates expectations B about conceptual structures.

Expectation uncertainty is represented by a probability distribution P(B) which is non-zero for a set of conceptual structures expected at a given time. Thus for a general concept structure  $\Gamma$  and a description Y of the speech signal, one gets:

$$\Gamma^* = \arg \max_{\Gamma} P(\Gamma | Y) = \arg \max_{\Gamma} P(\Gamma, Y)$$
$$P(\Gamma, Y) = \sum_B P(\Gamma, Y, B) \approx \max_B P(\Gamma, Y, B)$$

$$P(\Gamma, Y, B) = \sum_W P(W, \Gamma, Y, B) \approx$$
$$\max_W P(Y | W)P(W, \Gamma, B) \quad (1)$$
$$\Gamma^* \approx \arg \max_{\Gamma, W, B} \{P(Y | W)P(W, \Gamma, B)\}$$

$$P(\Gamma, W, B) = P(\Gamma | BW)P(W | B)P(B)$$

A general concept structure  $\Gamma$  can be represented as a string of parenthesized terminals and non-terminals.

These expressions can be decomposed into chunks. A sentence may contain only one or more chunks of an incomplete structure. Thus, a system should be able to generate interpretation hypotheses about parts of a conceptual structure. In this case, symbol  $\Gamma$  makes reference only to a set of components.

Probability  $P(\Gamma|BW)$  can be simply set equal to 0 for a conceptual structure which cannot be inferred from  $W$ . If the conceptual structure is part of the expectations of system beliefs and can be inferred unambiguously from  $W$ , then  $P(\Gamma|BW)$  as in many practical applications including the one considered in this paper, then  $P(\Gamma|BW)$ . Otherwise, let  $[c_1 \dots c_\gamma \dots c_\Gamma]$  be the sequence of concept symbols corresponding to the preterminal symbols in  $\Gamma$ . Probability  $P(\Gamma|BW)$  can be expressed as follows:

$$P(\Gamma | BW) = P([c_1 \dots c_\gamma \dots c_\Gamma] | BW) = P(c_1 | BW) \prod_{\gamma=2}^{\Gamma} P\{c_\gamma | [c_1 \dots c_{\gamma-1}] BW\} \quad (2)$$

At least, for some values of  $\gamma$  the probability  $P\{c_\gamma | [c_1 \dots c_{\gamma-1}] BW\}$  is one for a class of applications.

Let  $\Phi$  be the set of conceptual components, chunks of them or conceptual structures known to the system. Expectations derived from the system belief can be grouped into a set  $B1$ . Let  $B2$  the complement of  $B1$  w.r.t.  $\Phi$  and  $F$  be a filler structure representing all the conceptual structures not in the application or just ignored by ignorance of the system knowledge.  $B1$ ,  $B2$  and  $F$  are the possible values for  $B$  in the (1) and their probabilities  $P(B)$  can be established subjectively or by evaluating counts for user responses consistent with the belief, consistent with the application but not with the belief and inconsistent with the application knowledge.

Probability  $P(W|B)$  is that of an LM which is adapted to the system belief. It can be obtained with an LM built in the following way.

Each conceptual structure or part of it  $\Gamma$  is represented by a finite-state network  $N(\Gamma)$ .

All the networks corresponding to structures in  $B1$  are connected in parallel in a single structure with associated a probability  $P(B1)$ . A similar structure is built for the automata corresponding to structures in  $B2$ . A filler  $F$  is also considered containing a network derived by a trigram LM. A network  $N(\Gamma)$  is obtained by the concatenation of finite-state automata  $C(\Gamma)$  inferred with

the procedure described in the next section representing chunks of knowledge with fillers  $F$ . These automata output components of conceptual structures.

A search is performed by finding the most likely common path in the network and in the automaton derived from a lattice of word hypotheses generated by the speech recognizer with the generic trigram LM. System belief make vary the topology of the network by dynamically changing the composition of sets  $B1$  and  $B2$ . Network recompilation can be avoided by just putting all the  $N(\Gamma)$  in parallel and dynamically assigning each network of  $B1$  a probability :

$$P[N(\Gamma)] = \frac{P(B1)}{|B1|} \quad (3)$$

where  $|B1|$  indicates the number of elements in  $B1$ . Probabilities of networks in  $B2$  are assigned in a similar way.

A word sequence  $W$  always corresponds to a path in  $F$  and may correspond to one or more conceptual structures represented by paths in networks in  $B1$  and  $B2$ . In the second case, the likelihood of  $W$  in  $F$  will be much lower than the likelihood in  $B1$  or  $B2$  because phrases recognized by the chunk automata of the network are boosted as it will be shown later. Thus the best path for  $W$ , in this case, will go through a network whose automata produce as output the components of a conceptual structure.

### 3 Knowledge inference

Usually, when an application is developed, an even small training corpus is available.

Semantic categories and functions are manually derived for an application. They can be modified when the application is deployed in order to correct errors or add missing constituents.

A number of words in the lexicon have lexical entries containing their syntactic category, syntactic constructs which can appear in the same sentence, semantic features and constructs they can be part of. When one of these words is encountered in the training corpus, it is considered as a trigger for the semantic categories contained in its lexical entry. The association between words and semantic features is part of the *semantic knowledge* of the system.

The presence of a category in the sentence under analysis can be verified manually or by deriving it from the parse tree of the sentence. As lexical entries, grammars and rules for deriving semantic structures from parse

trees may be imprecise or incomplete, a single example can be carefully examined and validated manually.

Once a single example is available with a detailed syntactic and semantic analysis, it can be generalized. A sentence may contain a complete or partial semantic structure or just one component concept. Let  $\Gamma$  represent such a semantic interpretation. Furthermore, each structure may correspond to a pattern made of phrases and fillers of the sentence represented by a sequence of words  $W$ . Semantic Classification Trees (SCT) proposed in [1] can be used for automatically deriving sentence patterns corresponding to conceptual structures.

The purpose of learning is to build or modify a SFST that accepts a sequence of words and output a semantic interpretation  $\Gamma$ .

The initial analysis of an example starts by using a tagger for replacing words with their preterminal syntactic categories.

Then, semantic tags are automatically associated with sequences of syntactic tags manually or using the semantic knowledge. A *tag expression* made of syntactic and semantic tags is obtained in this way as a representation for of  $\Gamma$ . As a by-product, expressions for the constituents of and components of  $\Gamma$  are built and added to the semantic knowledge.

Generalization of the example uses a phrase generator to produce sequences of words from the tag expression. These sequences of words enrich the finite state translator which has to map word sequences into the conceptual structure  $\Gamma$ .

Further generalization can be obtained by inferring synonyms with a WordNet. If generalization has provided erroneous sequences of words, these sequences can be removed by manual inspection or when it is observed that the system has made an interpretation error because of them. With a similar procedure, new sequences of words can be added to the automaton for  $\Gamma$ .

Once it has been found that a word (noun or verb) contributed to hypothesize a concept in the semantic structure, the concept is added as semantic feature in the lexical entry of the word.

In summary learning of semantic knowledge follows the following steps:

1 Set the semantic categories for the application.

2 Set the lexical entries for the words that are semantically relevant for the application.

3 For every analyzed sentence

- if semantic interpretation is correct then do nothing,
- if a phrase is misplaced in the representation of a semantic structure then remove it,
- if a phrase is missed in the representation of a semantic structure, but the corresponding tag expressions is present in the semantic knowledge, then the phrase is added to the corresponding SFST,
- if the tag expression does not exist in the semantic knowledge, then it is built and sequences of words are generated from it with the above outlined generalization procedure.

A set of SFST is built in this way. They are added to the LM to provide concept specific components and to produce semantic interpretations at the same time with a translation process.

## References

- [1] Kuhn R. and De Mori R. (1995). The Application of Semantic Classification Trees to Natural Language Understanding. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17 : 449-460.
- [2] Pieraccini R., Levin E., and Lee C.-H. (1991). Stochastic Representation of Conceptual Structure in the ATIS Task. Proceedings of the, 1991 *Speech and Natural Language Workshop*, 121-124, Morgan Kaufmann publ, Los Altos, CA.
- [3] Vossen P. Diez-Orzas P. and Peters W., (1997) The multilingual design of EuroWordnet. Proc ACL/EACL workshop on automatic information extraction and building of lexical semantic resources for NLP applications, Madrid, 1997.