

Conceptual Decoding for Spoken Dialog systems

Yannick Estève, Christian Raymond, Frédéric Béchet,
Renato De Mori

LIA-CNRS
University of Avignon - France

{yannick.esteve, christian.raymond, frederic.bechet, renato.demori}@lia.univ-avignon.fr

Abstract

A search methodology is proposed for performing conceptual decoding process. Such a process provides the best sequence of word hypotheses according to a set of conceptual interpretations. The resulting models are combined in a network of Stochastic Finite State Transducers. This approach is a framework that tries to bridge the gap between speech recognition and speech understanding processes. Indeed, conceptual interpretations are generated according to both a semantic representation of the task and a system belief which evolves according to the dialogue states. Preliminary experiments on the detection of semantic entities (mainly named entities) in a dialog application have shown that interesting results can be obtained even if the Word Error Rate is pretty high.

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.

Integrating semantic concept models into a statistical Language Model (LM) is not a novel idea [7, 8, 9]. However, most of these studies use conceptual models either to rescore a n-best list of hypothesis or to semantically tag the best string output by the ASR module.

In our approach, the concepts are embedded in a set of global semantic interpretation of an utterance which evolves according to the state of the dialogue. Finding the best interpretation from a word graph output by the ASR module and a dialogue state leads to produce a word string as well as a sequence of concepts.

A semantic interpretation is represented by a Finite State Transducer (FST) encoding regular grammars for each kind of concepts and filler models for the background text.

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 and the ASR output. This process is represented by composition operations between several FST, the first one being a word graph output by the ASR module, and the others being the different LM and concept models.

2. Hypothesis evaluation and search

2.1. Statistical model

Let a dialogue system have a belief which generates expectations B about conceptual structures (or semantic interpretations). 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 G and a description Y of the speech signal, one gets:

$$\Gamma^* = \underset{\Gamma}{\operatorname{argmax}} P(\Gamma | Y) = \underset{\Gamma}{\operatorname{argmax}} P(\Gamma, Y)$$

$$P(\Gamma, Y) = \sum_B P(\Gamma, Y, B) \approx \max_B P(\Gamma, Y, B)$$

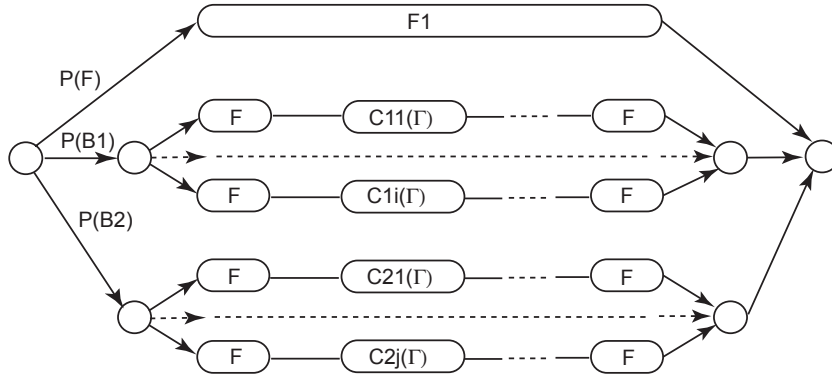


Figure 1: Network of integrated knowledge

$$\begin{aligned}
 P(\Gamma, Y, B) &= \sum_W P(W, \Gamma, Y, B) \\
 &\approx \max_W P(Y | W)P(W, \Gamma, B) \quad (1)
 \end{aligned}$$

$$\Gamma^* \approx \underset{\Gamma, W, B}{\operatorname{argmax}} \{P(Y | W)P(W, \Gamma, B)\}$$

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

A general concept structure G 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 Γ 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 $P(\Gamma | BW)$ can be set to 1 as in many practical applications including the one considered in this paper.

Let Φ be the set of conceptual components, chunks of them or conceptual structures or semantic interpretations 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. Φ 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 formula (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.

2.2. Building an integrated knowledge network

Each conceptual structure or part of it Γ is represented by a finite-state transducer $N(\Gamma)$. The input symbols of this transducer are the words and the output symbols are the concept

labels. 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 for accepting all the strings which are not parsed by $B1$ and $B2$. All these operations are done thanks to the AT&T FSM library presented in [5]. For example, removing from the filler all the paths accepted by $B1$ and $B2$ consists of simply applying an *fsmdifference* operation between the corresponding Finite State Machines (FSM).

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. The resulting network is represented in figure 1.

2.3. Search process

A search is performed by finding the most likely common path in the network $N(\Gamma)$ and in the automaton A derived from a lattice of word hypotheses generated by the speech recognizer. The automaton A is first composed with a bigram LM, also coded into an FSM (for more details about compositions between FSMs and language modeling, see [6]).

System belief makes vary the topology of the network $N(\Gamma)$ by dynamically changing the composition of sets $B1$ and $B2$. Network recompilation can be avoided by simply updating the probabilities between $B1$ and $B2$ without changing the topology of $N(\Gamma)$.

The result of the composition of A and the LM is then composed with $N(\Gamma)$ in order to obtain the final transducer R . Finding the best path in R consists of finding the best interpretation Γ according to equation (1). Let's point out that the FSM R can also be used in order to evaluate a specific interpretation proposed by the Dialogue Manager (DM). It can also output N -best lists on the concepts, instead of the words. This can be very interesting for the DM to obtain such lists as the usual N -best strings often differ only by some non-content words.

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 Γ 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 Γ .

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 of Γ . As a by-product, expressions for the constituents of and components of Γ 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 Γ .

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 Γ .

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 transducers is built in this way. They are used to provide concept specific components and to produce semantic interpretations at the same time with a translation process.

4. Experimental results

Some preliminary experiments were carried out on a dialog corpus provided by France Telecom R&D for tourist inquiries. The test corpus contains 1274 sentences collected over the French telephone network in different days. The task has a vocabulary of 2200 words.

A lattice of word hypotheses was obtained for each spoken sentence and transformed into an automaton A which was composed with a bigram LM and the network shown in figure 1.

The LM was obtained with SRILM [4], while the network composition and subtraction were performed with the AT&T FSM Library [5].

The semantic interpretations or conceptual structures contained five kind of concepts corresponding to semantic entities relevant to the application (location names, restaurant specialties, etc.).

The following types of errors were considered:

- Word Error Rate: WER.
- Concept Undetected: CU
- Concept Substitution: CS
- Incorrect Concept Inserted: CI.
- Sentence Interpretation Error: SIE

The results are reported in Table 1.

| WER | CU | CS | CI | SIE |
|-------|-----|------|-------|-----|
| 38.7% | 18% | 2.7% | 12.9% | 12% |

Table 1: Concept and Word Error Rates

These are preliminary results as the set of concepts and semantic interpretations used were very limited.

The results show that concept insertions often appear more than once in the same sentence.

It is worth mentioning that the insertion rate is only 3.9% for the sentences actually containing named entities. Probably, when all the conceptual entities will be used, a better balance of types of errors will be observed.

Most of the errors are due to undetected concepts. This suggests that the process of inferring conceptual models requires a better generalization.

A detailed analysis of types of errors suggests the following error classification in which the order reflects the contribution to the overall error rate:

- absence of semantically relevant words in the word lattice,
- all semantically relevant words are in the word lattice, but some of them have a very low acoustic score,
- errors due to Out Of Vocabulary (OOV) words,
- weak generalization of conceptual language models.

The first type is responsible for half of the errors. The second type accounts for 41.5% of the cases and the remaining errors are almost equally distributed among the last two.

5. Conclusion

A search methodology has been proposed for using conceptual models in a decoding process which provides at the same time the best sequence of word hypotheses according to a set of conceptual interpretations.

Preliminary experiments on the detection of semantic entities in a dialog application have shown that interesting results can be obtained even if the WER is pretty high.

Research will continue by progressively adding to the system knowledge models for all the conceptual structures and their fragments. Knowledge will continue to be integrated with the FSM Library [5] framework which appears to be a very powerful tool for combining models operating on different levels.

New linguistically and semantically based confidence criteria will be developed in order to provide the Dialog Manager with linguistic values expressing recognition and interpretation confidence.

6. Acknowledgements

This research was carried out with support from France Telecom R&D under Grant n° 021B178. The Authors are grateful to Denis Jovet, David Sadek, Frank Panaget, Céline Ancé and Géraldine Damnati.

The Authors also wish to thank the authors of the SRILM toolkit and the authors of the AT&T FSM Library for making available their tools.

7. 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.
- [4] Stolcke A. (2002). SRILM-An extensible language modeling toolkit Proc. *International Conference on Spoken Language Processing*, vol. 2, pp. 901-904, Denver, CO
- [5] Mohri M., Pereira F., Riley M. (2000). The design principles of a Weighted Finite-State Transducer Library. *Theoretical Computer Science*, 231, 17-32.
- [6] Mohri M., Pereira F., Riley M. (2002). Weighted Finite-State Transducer in Speech Recognition. *Computer Speech and Language* 16(1), 69-88
- [7] Hacioglu K., Ward W (2001). Dialog-Dependent Language Modeling Combining N-Grams And Stochastic Context Free Grammars, *ICASSP 2001*, Salt Lake City, USA
- [8] Erdogan H., Sarikaya R., Gao Y., Picheny M. (2002). Semantic Structured Language Models, *ICSLP 2002*, Denver, USA
- [9] Bonneau-Maynard H, Lefevre F. (2001). Investigating Stochastic Speech Understanding, *ASRU 2001*, Trento, Italy