ARTICLE IN PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Speech Communication xxx (2005) xxx–xxx

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

On the use of finite state transducers for semantic interpretation ☆

Christian Raymond [a], Frédéric Béchet [a,*],
Renato De Mori [a], Géraldine Damnati [b]

[a] LIA/CNRS, University of Avignon, BP1228, 84911 Avignon Cedex 09, France
[b] France Télécom R&D TECH/SSTP, 2 av. Pierre Marzin, 22307 Lannion Cedex 07, France

Received 19 January 2005; received in revised form 11 May 2005; accepted 24 June 2005

Abstract

A spoken language understanding (SLU) system is described. It generates hypotheses of conceptual constituents with a translation process. This process is performed by finite state transducers (FST) which accept word patterns from a lattice of word hypotheses generated by an Automatic Speech Recognition (ASR) system. FSTs operate in parallel and may share word hypotheses at their input. Semantic hypotheses are obtained by composition of compatible translations under the control of composition rules. Interpretation hypotheses are scored by the sum of the posterior probabilities of paths in the lattice of word hypotheses supporting the interpretation. A compact structured *n*-best list of interpretation is obtained and used by the SLU interpretation strategy.

Keywords: Automatic speech recognition; Spoken language understanding, Dialogue; Finite state transducers

1. Introduction

Semantic interpretations are data structures built using a knowledge source and performing operations such as recognition of conceptual constituents, composition and inference. Hypothesisation of conceptual constituents is triggered by words, phrases and other structures. Composition and inference generate semantic structures from conceptual constituents and other structures

☆ This research is supported by France Telecom's R&D under the contract 021B178.

* Corresponding author.

E-mail addresses: christian.raymond@univ-avignon.fr (C. Raymond), frederic.bechet@univ-avignon.fr (F. Béchet), renato.demori@univ-avignon.fr (R. De Mori), geraldine.damnati@rd.francetelecom.com (G. Damnati).

0167-6393/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.specom.2005.06.012

following different types of semantic relations between constituents. The interpretation of a message conveyed by a speech signal involves the generation of semantic hypothesis from word hypotheses.

Most of the approaches proposed so far for spoken language understanding (SLU) integrate semantic and syntactic knowledge into a semantic grammar and propose different algorithms for computing the probability $P(\Gamma, W)$ of a conceptual structure Γ and a sequence of words W (Sarikaya et al., 2004; Bangalore and Johnston, 2004; Wang et al., 2002; He and Young, 2003; He and Young, 2004; Seneff, 1992; Hacioglu and Ward, 2001; Young et al., 1989).

In contrast to this, approaches to natural language understanding (NLU) have been proposed with separate language and semantic models. Interesting comments and motivations for approaches of this type can be found in (Jackendoff, 1990). Among them, it is worth mentioning:

- Each major syntactic constituent of a sentence maps into a conceptual constituent, but the inverse is not true.
- Each conceptual constituent supports the encoding of units (linguistic, visual, etc.).
- Many of the categories support type/token distinction (e.g. place_type, place_token).
- Many of the categories support quantification.
- Each conceptual category has some realizations in which it is decomposed into a function/argument structure.
- Various types of relations, such as IS_A, PART_OF, hold between conceptual constituents. These relations can be used to infer the presence of a constituent in a sentence given the presence of other constituents.

Various types of semantic knowledge and their use can be found, for example in (Jackendoff, 1990; Levesque and Brachman, 1985). A common aspect of many of them is that it is possible to represent complex relational structures with non-probabilistic schemes that are more effective than context-free grammars. The purpose of this paper is to describe a process for generating hypotheses about a set of basic conceptual constituents. A conceptual language model (LM) is proposed for each constituent. These models taken into account long distance word dependencies and are implemented by finite state machines (FSM). A strategy is proposed for generating interpretation hypotheses by transforming a lattice of word hypotheses generated by an ASR system into a stochastic FSM and performing algebraic operations between this stochastic FSM and the FSMs implementing conceptual LMs.

Semantic composition and inference are not discussed in this paper. Nevertheless the formalism described in (Jackendoff, 1990) is used to show that the proposed approach is consistent with a linguistic theory rich of examples about the application domain of the system used for the experiment described later on. In particular, with the proposed strategy, it is possible that the same word contributes to the generation of different conceptual hypotheses. This makes it possible to introduce redundancies in conceptual LMs to increase hypothesisation robustness.

The detection and the extraction of the basic concepts from the speech signal will be described in Section 3 with a simple example. For semantic scoring and evaluation, it is convenient to estimate the probability of a semantic interpretation given the acoustic description of the speech signal, in contrast with approaches that consider an approximation which completely separates the likelihood of concepts and words from the likelihood of acoustic descriptions and words. Semantic composition is described in Section 4. Section 5 describes how a *structured n-best list* of interpretations is extracted from a word lattice generated by the ASR module processing a speech signal. This list contains *all* the interpretations that can be found in the word lattice, with their posterior probabilities, and the *n*-best values for each interpretation. Finally, the last part of this paper describes the integration of the proposed SLU module in a dialogue manager, involving a decision strategy based on various confidence measures.

2. Knowledge representation and use

The knowledge representation used in the approach proposed in the following is based on a two-level model: the first level is made of conceptual constituents each of which belongs to one of a small set of major ontological categories (such as `thing`, `event`, `state`, `action`, `place`, `path`, `property`, `amount`, `etc.`). Section 3 presents how these concepts are represented and extracted from a speech signal by means of specific LMs (one for each concept) coded as finite state machines (FSM). The second level in the knowledge representation consists of semantic relations between these conceptual constituents. This level is considered in this paper only because conceptual LMs have been derived only for few composite concepts. Composition has been manually performed following the theory and examples described in (Jackendoff, 1990). Nevertheless, this formalism is not used in the system implementation.

The theory is summarized as follows: Let *SPS* be an alphabet of major conceptual categories called semantic parts of speech. Each of them is represented in terms of functions and arguments. Let *FA* be the alphabet of the functions. Arguments are entities E_k which can be element of SPS or results of compositions. Composition rules are expressions of the type $[E_m] \rightarrow F([E_i],[E_j],[E_k],\ldots)$. The operators of $F \in FA$ are disjunction and conjunctions. An example of composition is:

$$[\text{PLACE}] \rightarrow \lfloor_{\text{place}} IN(\lfloor_{\text{thing}}\text{LOC}\rfloor)\rfloor$$

Subscripts are *ontological category variables*. `IN` indicates a *function* whose argument follows between parentheses. Selectional restrictions are general semantic restrictions on arguments. In the above example, `LOC` is a restriction for `THING`.

Semantic composition rules are also used to derive patterns of compatibility among hypotheses of conceptual constituents. A number of such patterns have been manually derived and used as described in Section 4.

In summary, structures are obtained with semantic knowledge. LMs and FSMs are designed for each basic structure. Interpretation hypotheses are generated with operations on automata. Compatibility among interpretation hypotheses is verified using semantic knowledge. Interpretation hypotheses are ranked using probabilistic scores.

3. Generation of elementary semantic hypotheses with a translation process

Reasoning with semantic knowledge is triggered by assertions of conceptual constituents which are elementary semantic hypotheses derived from recognized sentences or phrases which may not be syntactically well formed.

Semantic interpretation may follow syntactic analysis. The lexicon knowledge source has items in which words are associated with syntactic constraints and patterns of semantic structures. An instance of a pattern structure has to be built using the constituents of the sentence to be interpreted. Although noticeable examples exist on syntactic parsing and ASR (Roark, 2002; Chappelier et al., 1999), spontaneous speech in a dialogue context contains a lot of irregular syntactic forms, there may be errors in the hypothesised word sequences, lexical structures are incomplete and imprecise. Furthermore ASR accuracy can be improved by taking into account the fact that interpretation may constrain ASR and can be constrained by dialogue expectations or beliefs.

Following previous work from (Vidal et al., 1993), the above considerations suggest conceiving interpretation as a translation process in which stochastic language models (LMs) contain phrases accepted by finite state machines (FSM) which output labels for semantic constituents. There is an FSM for each elementary conceptual constituent. The definition of these constituents has to satisfy two major constraints:

- it should be possible to infer a LM for each constituent;
- it should be possible to compose or infer any structure of the semantic knowledge from the set of constituents.

The LM for each constituent can be seen as a language accepted by a finite state approximation of a natural language grammar. Such an approximation is implemented by a FSM transducer whose outputs are instances of semantic constituents.

This idea has been implemented in the past representing the linguistic knowledge of a conceptual constituent with a hidden Markov model (HMM) having words as observations (Levin and Pieraccini, 1995) or with a collection of word patterns made of chains of phrases and fillers inferred by semantic classification trees (SCTs) (Kuhn and De Mori, 1995). SCTs have also been the first example of the application of classifiers to semantic interpretation, a practice that is now fairly popular (Haffner et al., 2003). Previous speech understanding systems based on Finite State parsing can also be found in (Kaiser et al., 1999, Potamianos and Kuo, 2000).

Different automata, representing different conceptual constituents, may share phrases, allowing semantic interpretations to share part of the words conveyed by a speech signal, without requiring, as in popular approaches to shallow semantic parsing, that semantic chunks do not overlap (Hacioglu, 2004; Pradhan et al., 2004).

The structure of these FSMs is described with a simple example before introducing the decoding strategy. The dialogue corpus used for training and evaluating these algorithms is made of dialogue traces of the France Telecom R&D spoken dialogue system PlanResto (Sadek et al., 1996) which is a booking restaurant application. All the examples presented here are related to this application.

3.1. Representing conceptual constituents as FSMs

The conceptual constituents (called γ in this paper) that represent the first level in the interpretation strategy are represented by FSM transducers. These transducers take words at the input and output the label γ corresponding to the constituent conveyed by the accepted phrase. Their definitions rely on the dialogue strategy and they can be either related to dialogue management (confirmation, contestation, etc.) or to the application domain (location, date, etc.).

General entities like date, amount or phone numbers can be shared by several dialogue applications. They are represented by hand-written regular grammars coded as FSMs. For application-dependent entities, a lightly supervised method is used for automatically extracting the word string patterns on which the FSMs are induced as presented in (Béchet et al., 2002).

FSMs are induced from these patterns by generalizing some of their tokens by means of syntactic and semantic criteria (e.g. prepositions, digits or month names).

For example, from the utterance: *le deuxième restaurant italien* (*the second Italian restaurant*), two phrases can be extracted: the first one *le deuxième restaurant* in order to characterize the concept $\gamma = \texttt{access\_to\_a\_list}$ and the second one *restaurant italien* for characterizing the concept $\gamma = \texttt{restaurant}$. From these phrases, the patterns (`$ORD restaurant`) and (`restaurant $SPEC`) can be derived, with the non-terminal symbol `$ORD` representing any ordinal number and `$SPEC` any food speciality. These patterns are then turned into FSMs.

The fact that these two patterns share the word *restaurant* is not a problem with the proposed decoding method, as it will be shown in the next sections.

Let us point out that the only supervision needed in this process is the manual annotation of the corpus according to the concepts γ and the choice of the generalization rules.

3.2. FSM definitions and algorithms

All the operations presented on the FSMs are made with the AT&T FSM toolkit (Mohri et al., 1997). Following the definitions used in (Mohri et al., 2002) the acceptor and transducer FSMs used in this study are defined thanks to the general algebraic notion of *semiring*. A semiring K consists of a set $\mathbb{K}$ with an associative and commutative operation $\oplus$, an associative operation $\otimes$, as well as two identity elements: $\bar{0}$ and $\bar{1}$: $K = (\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$. The weights associated to the hypotheses generated by the ASR module represent probabilities implemented as $-\log$ probabilities. The corresponding semiring is called the *log semiring*: $(\mathbb{R}, +, ., 0, 1)$. When using $-\log$ probabilities with a best path approximation, the *tropical semiring* is used: $(\mathbb{R}_+ \cup \infty, \min, +, \infty, 0)$.

Acceptors and transducers are then defined as follows:

Let Σ be an alphabet of input symbols; Δ an alphabet of output symbols; ϵ an empty symbol; Q a set of states (with $I =$ initial states and $F =$

final states); $\mathbb{K}$ a semiring; E a set of transitions defined as: $E \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times (\Delta \cup \{\epsilon\}) \times \mathbb{K} \times Q$; w a weight function: w: $Q \rightarrow \mathbb{K}$.

If $Path(R_1, x, R_2)$ is a set of paths from $R_1 \subseteq Q$ to $R_2 \subseteq Q$ with input label x and $Path(R_1, x, y, R_2)$ a set of paths in $Path(R_1, x, R_2)$ with output label y, then:

- Acceptor $A = (\Sigma, Q, I, F, E)$ with for all $x \in \Sigma$:

$$[A](x) = \bigoplus_{\pi \in Path(I,x,F)} w[\pi] \tag{1}$$

- Transducer $T = (\Sigma, \Delta, Q, I, F, E)$ with for all $x \in \Sigma^*$, $y \in \Delta^*$:

$$[T](x,y) = \bigoplus_{\pi \in Path(I,x,y,F)} w[\pi] \tag{2}$$

and $w[\pi] = w[t_1] \otimes w[t_2] \otimes \cdots \otimes w[t_n]$ for a path π made of the following transitions $t_1, t_2, \ldots, t_n$.

In the next sections, the following fundamental operations on FSM are going to be used:

- Composition: $[T_1 \circ T_2](x,y) = \oplus_z [T_1](x,z) \otimes [T_2](z,y)$,
- Intersection: $[A_1 \cap A_2](x) = [A_1](x) \otimes [A_2](x)$,
- Difference: $[A_1 - A_2](x) = [A_1 \cap \overline{A_2}](x)$,
- Projection: $[A](x) = \oplus_y [T](x,y)$ and $[A](y) = \oplus_x [T](x,y)$.

3.3. Word-to-concept transducer

Each conceptual constituent $\gamma_k \in \Gamma$ is associated with an acceptor FSM (A_k for the concept γ_k). In order to process strings of words that do not belong to any concept, a filler model, called A_F is used. Because the same string of words cannot belong to both a concept model and the background text, all the paths contained in the acceptors A_k (noted: $A_\Gamma = \cup_{\gamma_k \in \Gamma} A_k$) are removed from the filler model A_F in order to produce the acceptor $A'_F : [A'_F](x) = [A_F - A_\Gamma](x)$.

All these acceptors are then turned into transducers that take words as input symbols and `start` or `end` concept tags as output symbols. Indeed, all acceptors A_k become transducers T_k where the first transition emits the symbol $\langle \gamma_k \rangle$ and the last transition the symbol $\langle / \gamma_k \rangle$. Similarly the filler model becomes the transducer T_{BK} which emits the symbols ⟨BAK⟩ and ⟨/BAK⟩. Except these start and end tags, no other symbols are emitted: all words in the concept or background transducers emit an empty ϵ symbol (coded `0` in the example figures).

Finally all these transducers are linked together in a single model called $T_{concept}$ as presented in Fig. 1.

3.4. Decoding process

The decoding process leading to the output of a n-best list of concept strings is described here and illustrated on a small example. When processing an utterance, the ASR module of the dialogue system outputs a word lattice coded as an acceptor G_W. The semiring used is the *log semiring* and the weigh function $w(\pi)$ corresponds to the log of the score $P(Y|W)P(W)$ where Y is a sequence of acoustic observations, W the string of words representing the path π, $P(Y|W)$ the probability given by the acoustic models and $P(W)$ the probability given by a bigram language model. An example of G_W is given in Fig. 2. The likelihood scores attached to each transition are given in $-logprob$ values.

G_W is composed with the transducer $T_{concept}$ in order to obtain the word-to-concept transducer T_{WC}:$T_{WC} = G_W \circ T_{concept}$, illustrated by Fig. 3.

In this example we consider four kind of conceptual constituents: the functions `NEAR` (for a location) and `LESS` (for an amount) and the tokens `LOC` for location and `AMOUNT` for monetary values. These constituents are represented by the following patterns in this example:

```
NEAR = pas loin du metro (near metro)
NEAR = pas loin du $NAME (near $NAME)
LOC = Trocadero
LOC = metro Opera
AMOUNT = $NUMBER euros
LESS = moins de $NUMBER (less than
$NUMBER)
```

with `$NAME` being any proper name and `$NUMBER` any number expression.

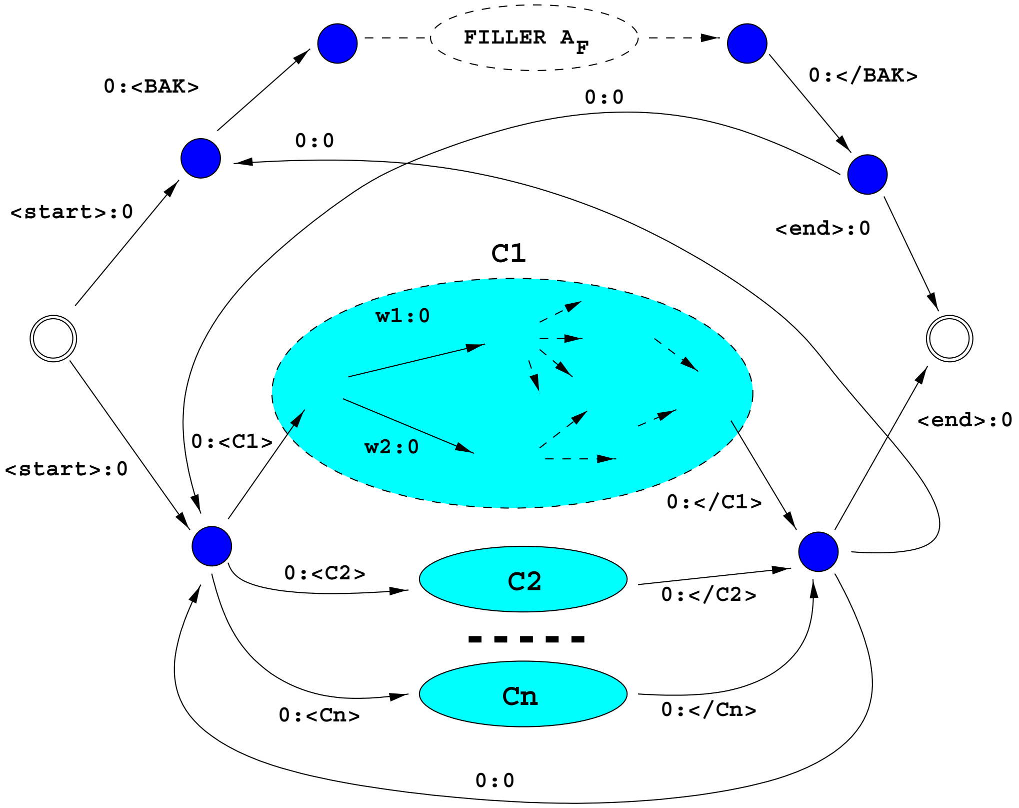

Fig. 1. Word-to-concept transducer T_{concept}.

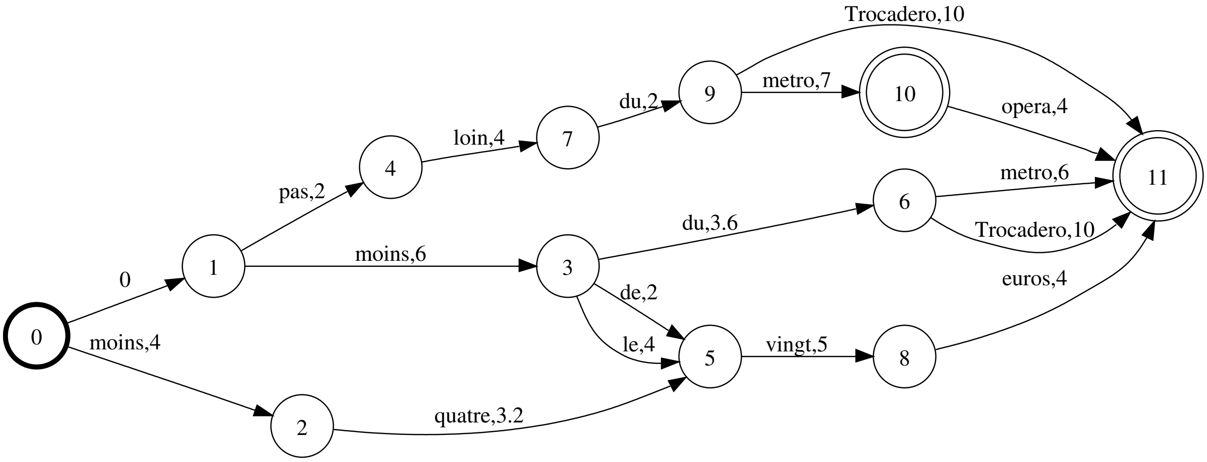

Fig. 2. Example of word lattice G_W output by an ASR module.

A path $Path(I, x, y, F)$ in T_{WC} (with I the initial state and F a final state of T_{WC}) is either a word string if one considers only the input symbols x or a concept string if one considers the output symbols y. In order to obtain all the possible interpretations contained in G_W, we project T_{WC} on the

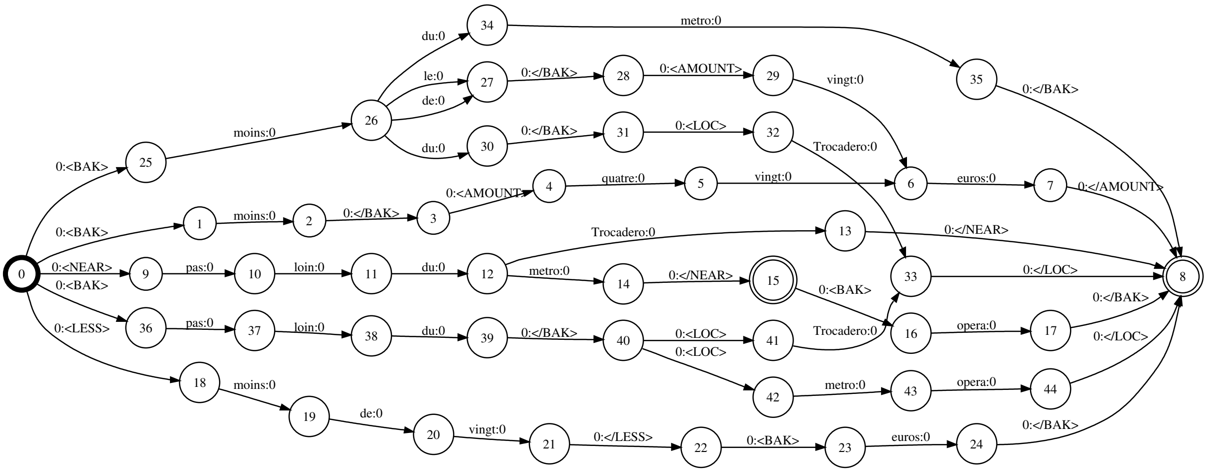

Fig. 3. Example of transducer T_{WC} corresponding to the composition of an acceptor representing a word lattice output by the ASR module and the word-to-concept transducer T_{concept}.

output symbols, then determinize and minimize the resulting FSM. The acceptor obtained is called G_{C}. Because this operation is done on the log semiring, the weigh of the path $Path(I, y, F)$ is the sum of the weights of all the paths in T_{WC} producing y:

$$[G_{\mathrm{C}}](y) = \bigoplus_{x} [T_{\mathrm{WC}}](x, y) \tag{3}$$

An example of G_{C}, obtained from the transducer T_{WC} of Fig. 3 is shown in Fig. 4. For clarity we omitted in this figure the end tags of the concepts. The double circled states are the final states.

The n-best list of conceptual interpretations $I_1, I_2, \ldots, I_n$ is obtained by first changing the semiring in G_{C}, from log semiring to tropical semiring, then enumerating the n best paths $S_1, S_2, \ldots, S_n$ in G_{C}. Each interpretation I_i is a string of labels γ represented by an acceptor S_i with the vocabulary in Γ. To each I_i is also attached an acceptor $G_{\mathrm{W}i}$ which is the set of paths in T_{WC} that output the string I_i: $[G_{\mathrm{W}_i}] = [T_{\mathrm{WC}} \circ S_i]$.

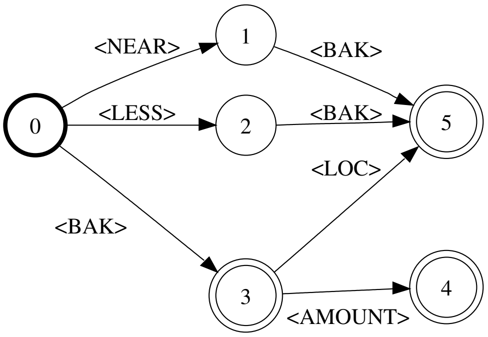

Fig. 4. Example of acceptor G_{C} obtained by projecting the transducer T_{WC} on the output symbols.

This is represented in Fig. 5.

This set of concept strings is the first step in the generation process of the *structured n-best* list of hypotheses $S_{n\text{-best}}$ which is the final output of our understanding module.

Notice that phrases for different concepts may overlap. In fact, the words expressing the meaning of a function may be essential or very useful for assessing that a proper name indicates unambiguously a particular type of location, for example, rather than another type of entity. A certain level of redundancy may also be very useful for compensating ASR errors. Because of this overlap some basic concepts (like NEAR and LOC in our example) are in different interpretations (I_1 and I_2 in Fig. 5). It is the second level in our understanding process that is going to merge, if possible, these concepts by applying semantic relations, as presented in the next section.

Let us point out also that this process has a negligible impact on the overall speed of the decoding because the operations on automata and transducers are very fast.

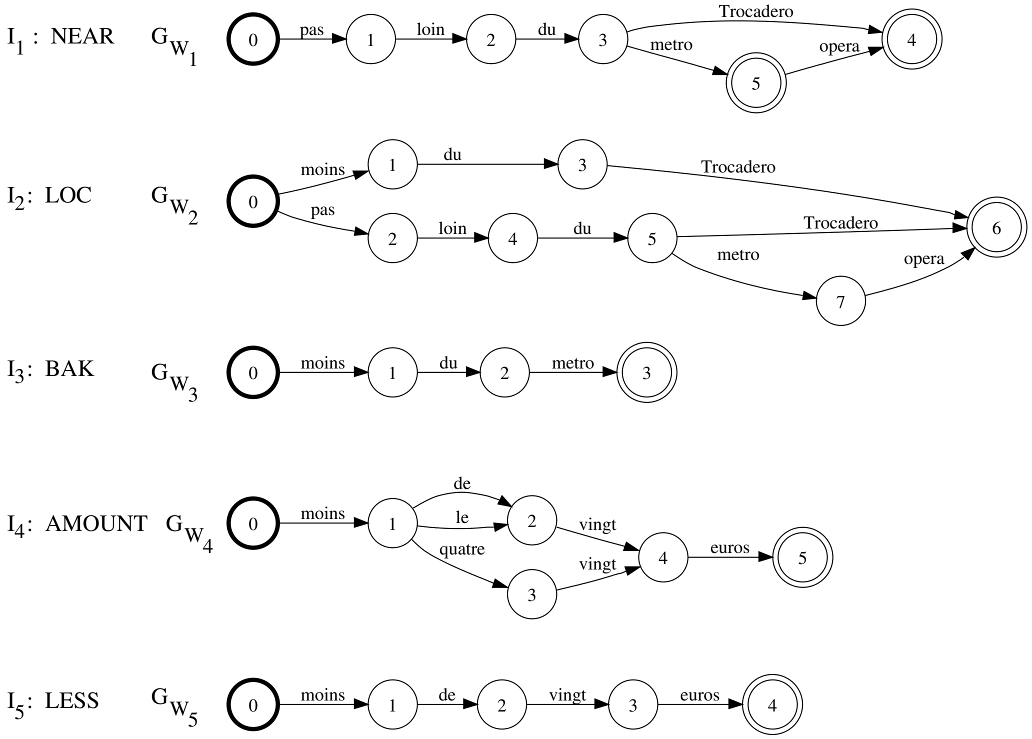

Fig. 5. *n*-Best list of basic conceptual interpretations I_i with their corresponding acceptor G_{W_i}.

4. Application of semantic relations

Once conceptual constituents have been hypothesised for a spoken sentence, semantic relations which apply to them have to be identified and used to instantiate composite semantic structures and to perform inferences. Composition is itself a type of inference with which compounds are inferred based on premises represented by components. It is worth recalling that relations may have a support in the lattice of word hypotheses and the result of an inference has a support too. This support is the intersection of the supports of all the premises. A support is always the union of paths spanning the entire sentence. An inference may not imply any temporal relation between premises.

Part of the semantic knowledge is made of implications expressing specialized formation rules. For the sake of clarity, an example of inference using these rules will be developed in this section. Using the notation proposed in (Jackendoff, 1990) the category path is inferred by the following rule:

$$[\text{PATH}] \rightarrow \begin{bmatrix} \left\{\begin{matrix} \text{TO}\\ \text{FROM}\\ \text{NEAR}\\ \text{TOWARD}\\ \ldots \end{matrix}\right\} \left(\left[\left\{\begin{matrix}\text{THING}\\ \text{PLACE}\end{matrix}\right\}\right]\right) \\ \text{path} \end{bmatrix}$$

The rule establishes that, for example, the composition of the function NEAR with an instance of PLACE results in an instance of PATH. Every composition obtained with formation rules is a well formed semantic structure. Composition rules can be used in more complex structured semantic representations such as KLONE (Brachman and Schmolze, 1985).

By inference, an instance of <PATH>: $\lfloor_{\text{path}}\text{NEAR}(\lfloor_{\text{place}}\text{IN}(\lfloor_{\text{thing}}\text{LOC}\rfloor)\rfloor)\rfloor$can be hypothesised by the presence of hypotheses of the function NEAR and an instance of <PLACE>.

If the hypothesis NEAR is represented with the acceptor $G_{\mathrm{W}_{\text{NEAR}}}$ and <PLACE> by the acceptor $G_{\mathrm{W}_{\text{PLACE}}}$ (obtained with the method presented in the previous section), then the hypothesis $G_{W_{\text{PATH}}}$ about an instance of <PATH> is generated if and only if

$$G_{\mathrm{W}_{\text{PATH}}} = [G_{\mathrm{W}_{\text{NEAR}}} \cap G_{\mathrm{W}_{\text{PLACE}}}] \neq \emptyset$$

If the newly generated semantic structures make applicable new formation rules, the supports of the corresponding constituents are intersected and, if the intersection is not empty, added to the set of interpretations. The process is repeated until no more compounds can be formed.

Each semantic structure I_i represented by the acceptor $G_{\mathrm{W}_{I_i}}$ is scored by the following posterior probability (with X the set of strings accepted by $G_{\mathrm{W}_{I}i}$ and Z the set of strings accepted by W_G):

$$P(I_i|Y) = \frac{\bigoplus_{x\in X}[G_{\mathrm{W}_{I_i}}](x)}{\bigoplus_{z\in Z}[W_G](z)} \qquad (4)$$

For example, from the five interpretations of Fig. 5, two semantic relations can be applied:

- one for <PLACE>, as presented above, based on the concepts NEAR (I_1) and LOC (I_2);

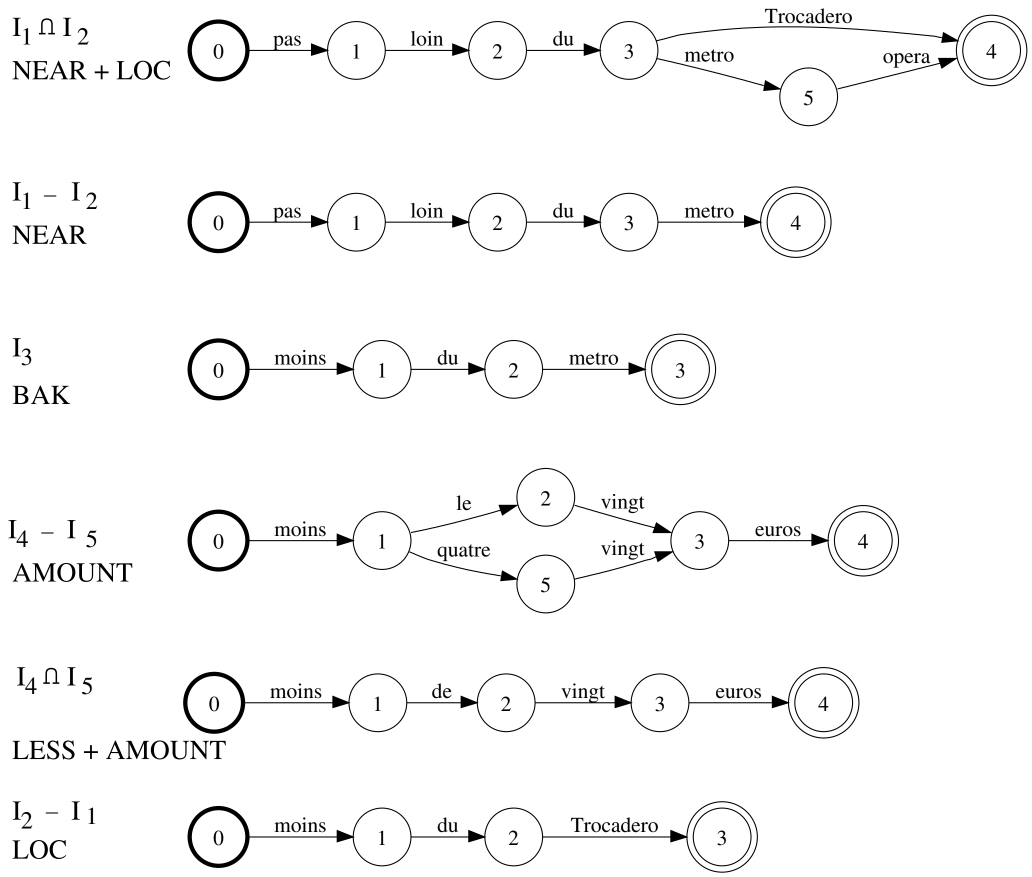

Fig. 6. *n*-Best list of interpretations (with their corresponding acceptor) after application of semantic relations to the *n*-best list of Fig. 5.

- one for the instance <MONEY>: $\lfloor_{\text{money}}\text{LESS}(\lfloor_{\text{thing}}\text{AMOUNT}\rfloor)\rfloor$ based on the concepts LESS (I_5) and AMOUNT (I_4).

This leads to perform 6 operations on the acceptors G_{W_i}:

$$G_{\text{W}_{I_1 \cap I_2}} = [G_{\text{W}_1} \cap G_{\text{W}_2}] \rightarrow \text{NEAR} + \text{LOC}$$
$$G_{\text{W}_{I_1 - I_2}} = [G_{\text{W}_1} - G_{\text{W}_2}] \rightarrow \text{NEAR}$$
$$G_{\text{W}_{I_2 - I_1}} = [G_{\text{W}_2} - G_{\text{W}_1}] \rightarrow \text{LOC}$$
$$G_{\text{W}_{I_4 \cap I_5}} = [G_{\text{W}_4} \cap G_{\text{W}_5}] \rightarrow \text{AMOUNT} + \text{LESS}$$
$$G_{\text{W}_{I_4 - I_5}} = [G_{\text{W}_4} - G_{\text{W}_5}] \rightarrow \text{AMOUNT}$$
$$G_{\text{W}_{I_5 - I_4}} = [G_{\text{W}_5} - G_{\text{W}_4}] \rightarrow \text{LESS}$$

Because $G_{\text{W}_{I_5 - I_4}} = \emptyset$ only six interpretations are kept: the 5 ones just obtained plus the interpretation I_3 (BAK) which have not been involved in any operations. These interpretations are scored according to their posterior probabilities. On the FSM of Fig. 2, we obtain:

$$P(I_1 \cap I_2) = 0.58 \quad P(I_1 - I_2) = 0.21$$
$$P(I_3) = 0.11 \quad P(I_4 - I_5) = 0.07$$
$$P(I_4 \cap I_5) = 0.028 \quad P(I_2 - I_1) = 0.002$$

Fig. 6 shows the six interpretations kept with their corresponding FSMs. Another method for scoring semantic interpretations will be introduced in Section 6.

5. Structured *n*-best list of semantic interpretations

The last step in the understanding process generates the *n*-best hypothesis list of *concept values* for each concept hypothesis. Several values can be found in an FSM for the same concept. This is particularly true when the concepts represent numerical entities like phone numbers or amounts. It is therefore possible to output not only the best word string for each interpretation I_i obtained but rather the *n*-best list of word strings leading to different concept values.

Extracting the *n*-best values can be very useful in a dialogue context, as some additional information (customer data, constraints on the values, etc.) can be used to select a value among a list of hypotheses. For example, Rahim et al. (2001) shows that using a general phone directory for filtering phone number *n*-best lists automatically extracted is a very efficient filter: the understanding accuracy of the phone strings that belong to the directory is 94.5% (and this represents 61% of the hypotheses) compared to only 45% accuracy for those that cannot be found in any phone directory.

Therefore, the acceptor $G_{\text{W}i}$ attached to each interpretation I_i is composed with a transducer that output only the different values (mainly proper names and numerical values) contained in the FSM. The *n*-best value strings as well as the corresponding support word strings are then associated to each I_i in order to build the *structured n-best list* of semantic interpretations ($S_{n\text{-best}}$).

The structured *n*-best list corresponding to our example is displayed in Table 1.

This kind of structure can be seen as an abstraction of all the possible interpretations of an utter-

Table 1
Example of structured *n*-best list obtained on the word lattice of Fig. 2 scored with their posterior probabilities

Rank	Interpretation/value	Score
I_1	$\lfloor_{\text{path}}\text{NEAR}(\lfloor_{\text{place}}IN(\lfloor_{\text{thing}}\text{LOC}\rfloor)\rfloor)\rfloor$	0.58
$I_{1.1}$	LOC(*type*: *subway*, *value* = *opera*)	0.57
W	pas loin du metro opera (near metro opera)	
$I_{1.2}$	LOC(*type*: *square*, *value* = *Trocadero*)	0.01
W	pas loin du Trocadero (near the Trocadero)	
I_2	NEAR	0.21
W	pas loin du metro (near metro)	
I_3	BAK	0.11
W	moins du metro (less metro)	
I_4	$\lfloor_{\text{thing}}\text{AMOUNT}\rfloor$	0.07
$I_{4.1}$	AMOUNT(*type*: *euros*, *value* = 80)	0.065
W	moins quatre vingt euros (less eighty euros)	
$I_{4.2}$	AMOUNT(*type*: *euros*, *value* = 20)	0.005
W	moins le vingt euros (less the twenty euros)	
I_5	$\lfloor_{\text{money}}\text{LESS}(\lfloor_{\text{thing}}\text{AMOUNT}\rfloor)\rfloor$	0.028
$I_{5.1}$	AMOUNT(*type*: *euros*, *value* = 20)	0.028
W	moins de vingt euros (less than twenty euros)	
I_6	$\lfloor_{\text{thing}}\text{LOC}\rfloor$	0.002
$I_{6.1}$	LOC(*type*: *square*, *value* = *Trocadero*)	0.002
W	moins du Trocadero (less the Trocadero)	

ance. The two main advantages of the semantically structured n-best list compared to standard n-best list of word strings are as follows:

1. Standard n-best lists are produced by enumerating the n-best paths produced by the ASR module. The scores used are a combination of acoustic and language model scores; no other linguistic level is involved. When an n-best word hypothesis list is generated, the differences between the hypothesis i and the hypothesis $i+1$ are often very small, made of only one or a few words. This phenomenon is aggravated when the ASR word lattice contains a low confidence segment, due for example to an out-of-vocabulary word, to a noisy input or to a speech disfluency. On the contrary, in a Spoken Dialogue context, not all words are important to the Dialogue Manager while the difference between the words in the list may be semantically irrelevant. A particular advantage of using a structured n-best list $S_{n\text{-best}}$ is that every hypothesis has a different meaning for the dialogue manager.
2. By having confidence scores (the posterior probabilities) for both interpretations at the concept level and interpretations with concept values, S_{nbest} can be used by the dialogue manager for answering two kinds of questions:
   - are the concepts expected according to the dialogue state in the list of the possible interpretations of an utterance, and with which confidence scores?
   - for a given concept, what are the possible values that can be found and with which confidence scores?

6. Rescoring of semantic structure hypotheses

In order to select a particular interpretation Γ from the structured n-best list, it is interesting to compute the probability $P(\Gamma|M)$ that Γ is correct, given a set of confidence measures M. The choice of the confidence measures determines the quality of the decision strategy. The measures used in this study are briefly presented in the next sections.

6.1. Confidence measures

6.1.1. Acoustic confidence measure (AC)

This confidence measure relies on the comparison of the acoustic likelihood provided by the speech recognition model for a given hypothesis to the one that would be provided by a totally unconstrained phoneme loop model. In order to be consistent with the general model, the acoustic units are kept identical and the loop is over context-dependent phonemes. This confidence measure is used at the utterance level and at the concept level (see Raymond et al. (2003) for more details).

6.1.2. Linguistic confidence measure (LC)

In order to assess the impact of the absence of observed trigrams as a potential cause of recognition errors, a Language Model consistency measure is introduced. This measure, inspired by measures proposed in (Estève et al., 2003), is simply, for a given word string candidate, the ratio between the number of trigrams observed in the training corpus of the language model vs. the total number of trigrams in the same word string. Its computation is very fast and the confidence scores obtained from it give interesting results as presented in (Estève et al., 2003).

6.1.3. Semantic confidence measure (SC)

Several studies have shown that text classification tools (like Support Vector Machines or Boosting algorithms) can be an efficient way of labeling an utterance transcription with a semantic label such as a call-type (Haffner et al., 2003) in a Spoken Dialogue context. In our case, the semantic labels attached to an utterance are the different concepts handled by the Dialogue Manager. One classifier is trained for each concept tag in the following way.

Each utterance of a training corpus is labeled with a tag, manually checked, indicating if a given concept occurs or not in the utterance. In order to let the classifier model the context of occurrence of a concept rather than its value we removed most of the concept headwords from the list of criterion used by the classifier. We also added to the training corpora the automatic transcriptions of the utterances in order to increase the robustness of the classifier to noisy data output by the ASR engine.

During the decision process, if the interpretation evaluated contains two concepts c_1 and c_2, then the classifiers corresponding to c_1 and c_2 are used to give to the utterance a confidence score of containing these two concepts.

The text classifier used in the experimental section is a decision-tree classifier based on the semantic classification trees introduced for the ATIS task by Kuhn and De Mori (1995) and used for semantic disambiguation in (Béchet et al., 2000).

6.1.4. Rank and posterior probability confidence measures (R, PP)

To the previous confidence measures we added the confidence scores attached to the structured n-best lists:

- the rank of each candidate in the n-best. This rank contains two numbers: the rank of the interpretation of the utterance and the rank of the utterance among those having the same interpretation.
- the posterior probability estimated on the word lattice for each interpretation and utterance.

6.1.5. Dialogue context confidence measure (DC)

The dialogue context represents the focus of the ongoing dialogue. Concepts can be in the system focus (or *belief*) or outside it in case of misrecognition from the system or context switch from the user. However, in-focus concepts are more likely to occur than the others. In order to use this information, the dialogue context is attached to each utterance of our training corpus by means of labels that indicate the kind of concepts expected in the following turn. A priori distribution of concepts according to dialogue context labels are estimated on the training corpus and used as confidence measures during the decoding process.

6.2. Decision tree based strategy

As the statistical dependencies between confidence measures are difficult to establish, their values are transformed into symbols by vector quantization (VQ) and conjunctions of these symbols expressing relevant statistical dependencies are represented by a decision tree trained with a development set of examples.

To train this tree a corpus is built where each example is an interpretation Γ extracted from the structured n-best lists of hypotheses S_{nbest} obtained on a development corpus of utterances. Each interpretation Γ is represented by the following features:

- a set M of discredited values corresponding to the confidence measures presented in the previous section;
- a label, which is either *correct* (Γ^c) if Γ contains no errors according to the reference corpus, or *incorrect* (Γ^e) otherwise.

The decision tree is trained in order to minimize the impurity of the distribution of the correct and incorrect examples after expanding a tree node. The questions attached to each node are related to the symbols expressing the various confidence measures describing each example. This process stops when no further drop in impurity can be achieved or when the size of the set of samples attached to a node is below a given threshold. At the end of the training process, a score consisting in the proportion of positive examples is associated to a leaf. This proportion is an estimation of the probability that the interpretation of a sentence leading to that leaf is correct. This probability is conditioned by the function of confidence measures which is the conjunction of the symbols associated to the branches of the path from the root to the leaf:

$$P(\Gamma^c|M) = \frac{P(\Gamma^c, M)}{P(\Gamma^c, M) + P(\Gamma^e, M)} \tag{5}$$

At each leaf of the tree, this probability $P(\Gamma^c|M)$ is computed and a decision of acceptance can be made if it is above a given threshold.

7. Dialogue strategy

Once concepts have been hypothesised, a dialogue system has to decide what action to perform.

Let $A = a_j$ be the set of actions a system can perform. Some of them can be requests for clarification or repetition. In particular, the system may request the repetition of the entire utterance. Performing an action has a certain risk and the decision about the action to perform has to be the one that minimizes the risk of user dissatisfaction.

It is thus possible that some or all the hypothesised components of a conceptual structure Γ do not correspond to the user intention because the word sequence W based on which the conceptual hypothesis has been generated contains some errors. In particular, there are requests for clarification or repetition which should be performed right after the interpretation of an utterance in order to reduce the stress of the user. It is important to notice that actions consisting in requests for clarification or repetition mostly depend on the probability that the interpretation of an utterance is correct, rather than on the utterance interpretation.

The decoding process described in Section 3.4 provides a number of hypotheses scored with their posterior probabilities. To each hypotheses is attached the set of confidence measures M. $P(\Gamma^c|M)$ is computed for these hypotheses as presented in the previous section. The results can be used to decide to accept an interpretation or to formulate a clarification question which may imply more hypotheses.

For the sake of simplicity only two actions are considered: accepting the hypothesis with the highest $P(\Gamma^c|M)$ or rejecting it. The risk associated to the acceptation decision is called ρ_{fa} and corresponds to the cost of a false acceptation of an incorrect interpretation. Similarly the risk associated to the rejection decision is called ρ_{fr} and corresponds to the cost of a false rejection of a correct interpretation. In a spoken dialogue context, ρ_{fa} is supposed to be higher than ρ_{fr}.

The choice of the action to perform is determined by a threshold δ on $P(\Gamma^c|M)$. This threshold is tuned on a development corpus by minimizing the total risk R expressed as follows:

$$R = \rho_{\text{fa}} \times \frac{N_{\text{fa}}}{N_{\text{total}}} + \rho_{\text{fr}} \times \frac{N_{\text{fr}}}{N_{\text{total}}} \tag{6}$$

N_{fa} and N_{fr} are the numbers of false acceptation and false rejection decisions on the development corpus for a given value of δ. N_{total} is the total number of examples available for tuning the strategy.

The final goal of the strategy is to make negligible N_{fa} and the best set of confidence measures is the one that minimizes N_{fr}. In fact, the cost of these cases is lower because the corresponding action has to be a request for repetition.

Instead of simply discarding an utterance if $P(\Gamma^c|M)$ is below δ, another strategy is investigated which consists of estimating the probability that the conceptual interpretation alone (without the concept values) is correct. This probability can be estimated the same way as $P(\Gamma^c|M)$ and can be used to choose a third kind of actions: accepting the conceptual meaning of an utterance but asking for clarifications about the values of the concepts.

A final decision about the strategy to be adopted should be based on statistics on system performance to be collected and updated after deploying the system on the telephone network.

8. Experiments

Experiments are carried out on a dialogue corpus provided by France Telecom R&D and collected for a tourism telephone service (Sadek et al., 1996). Two corpora, extracted from this dialogue corpus, are used in this study:

- a development corpus, made of 2.1k utterances, containing for each utterance the word lattice output by the France Telecom speech recognizer, the best word string in the lattice with an average word error rate (WER) of 25.8 and the reference transcription with the list of concepts occurring in it;
- a test corpus, made of 1.7k utterances, containing also word lattices, best word strings with an average WER of 27.0 and the reference transcriptions with conceptual information.

The development corpus is used to train the decision tree used in the decision module as presented in Section 6.2.

The task has a vocabulary of 2200 words.

The results are given according to the understanding error rate (UER) measure. This measure is related to the normalized values of the concepts detected. These values are obtained thanks to a set of rules that translate the word strings detected as concepts into tokens representing the values. To each concept is associated a single value.

With Γ being the interpretation of an utterance of the test corpus, UER is defined as follows:

$$\mathrm{UER} = \frac{S_c + S_v + D_c + I_c}{T} \times 100 \tag{7}$$

where S_c indicates the substitution of an attribute of Γ, S_v the substitution of a concept value, D_c indicates deletion of an attribute and I_c indicates insertion. T is the total number of concepts in the reference corpus.

Fifteen concept tags are used in these experiments.

8.1. Comparison between standard vs. structured *n*-best lists of hypotheses

Fig. 7 shows the comparison between a standard and a structured *n*-best list S_{nbest} obtained on our test corpus. This curve shows the lowest understanding error rate (UER) that can be found in a *n*-best list of word hypotheses. This measure is called the *Oracle* error rate. It is defined as follows, for a list L of N hypotheses $H_1, H_2, \ldots, H_N$:

$$Oracle(L) = \min_{1<=i<=N} \mathrm{UER}(H_i) \tag{8}$$

Different values of n are compared. In this experiment the lowest bound that can be reached is the oracle value of the whole word lattice, and its value is 4%. As we can see, S_{nbest} outperforms significantly the standard *n*-best list: for $n = 5$ the oracle value for S_{nbest} is already near the lowest bound, at 4.5%, while the standard *n*-best reaches the same value for $n = 35$. This results shows that the average length of the structured *n*-best lists is quite low (around five hypotheses). S_{nbest} is an effective summary of all the information contained in the word lattice output by the ASR module.

It is interesting to plot the relative error rate reduction obtained by manually choosing the Oracle hypothesis at the understanding (UER) and at the word (WER) level. Fig. 8 shows such curves. At $n = 5$, a 60% relative error rate reduction is obtained at the understanding level with S_{nbest} (compared to 40% with the standard list) but this leads to only 6.5% relative reduction in WER. It is clear here that the WER measure is not an adequate

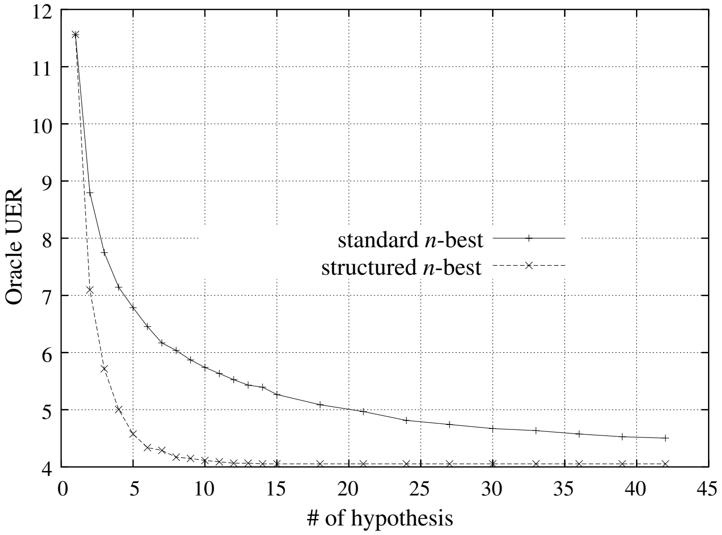

Fig. 7. Comparison of the lowest understanding error rate (Oracle UER) in a standard *n*-best list of hypotheses and a *n*-best list structured according to the utterance interpretations.

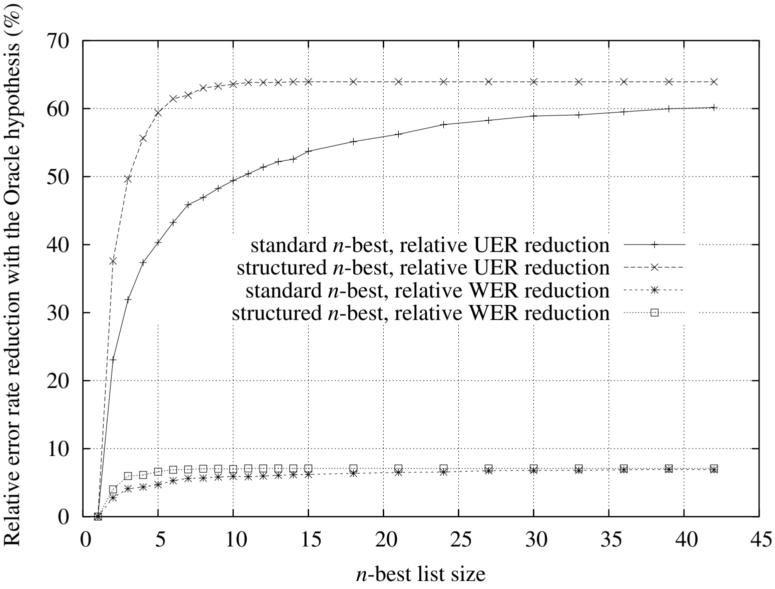

Fig. 8. Comparison of the relative error reduction for the understanding error rate and the word error rate obtained by manually choosing the Oracle hypothesis in a standard and a structured *n*-best list.

measure in a Spoken Dialogue context as a significant reduction in the Understanding Error Rate might have very little impact on the word error rate.

8.2. Evaluation of the rescoring strategy

Table 2 shows the results obtained with a rescoring strategy that selects, from S_{nbest}, the hypothesis with the highest $P(\Gamma^c|M)$ (M being a set of confidence measures). The baseline results are obtained with a standard maximum-likelihood approach choosing the hypothesis maximizing the probability $P(\Gamma|Y)$. No rejection is performed in this experiment and the size of the structured *n*-best list was limited to 10 interpretations.

Table 2
Understanding error rate results with and without rescoring on structured *n*-best lists

Corpus	Baseline	Rescoring	UER reduction %
Devt.	10.9	8.4	22.9
Test	11.3	9.7	14.2

The gain obtained after rescoring is significant and justifies the 2-step approach that first extract an *n*-best list of interpretations based on $P(\Gamma|Y)$ and then choose the one with the highest confidence according to a large set of confidence measures M.

8.3. Evaluation of the decision strategy

In this experiment we evaluate the decision strategy consisting of accepting or rejecting an hypothesis Γ thanks to a threshold on the probability $P(\Gamma^c|M)$. Fig. 9 shows the curve UER vs. utterance rejection on the test corpus. As we can see very significant improvements can be achieved with very little utterance rejection. For example, at a 10% utterance rejection operating point, the UER on the development corpus drops from 9.7% to 7.2% which is a 25% relative improvement.

By using Eq. (6) for finding the operating point minimizing the risk function (with a cost $\rho_{\text{fa}} = 1.5 \times \rho_{\text{fr}}$) on the development corpus we obtain an UER of 7.8% for a rejection rate of about 5%.

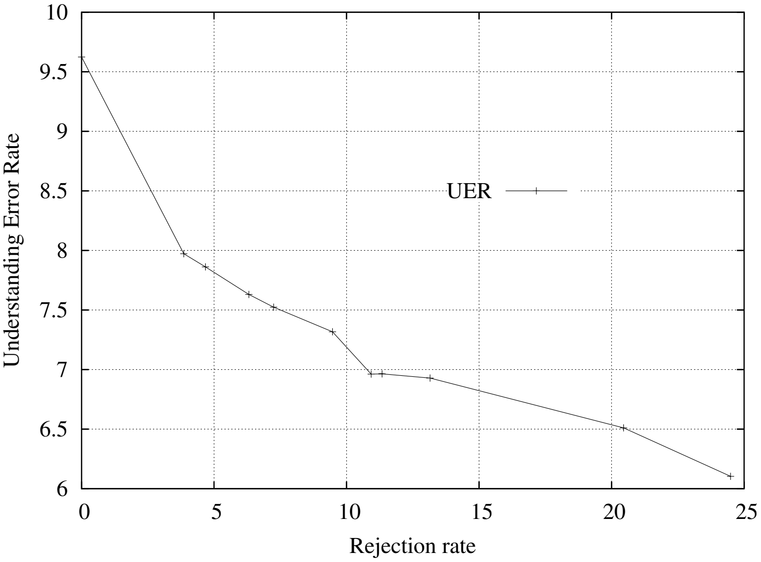

Fig. 9. Understanding error rate vs. utterance rejection on test corpus.

9. Conclusion

An effective use of FSMs for semantic interpretation has been described. A decoding algorithm has been proposed which uses FSMs to obtain a structured *n*-best list of sentence interpretation. This list is a short summary of the possible semantic contents of a spoken sentence. It is shown that comparable interpretation performance can be obtained with a structured list whose size is practically an order of magnitude lower than the use of the corresponding standard *n*-best list. This simplification significantly reduces the probability of interpretation errors. Strategies for selecting an interpretation in the structured list have been presented using confidence measures. Automatically trained decision trees are used to represent statistical dependences. Posterior probabilities that an interpretation is correct given a set of confidence measures are also used in a clarification dialogue strategy.

As FSM correspond to basic semantic constituents, they can be composed in various ways to generate a large number of semantic structures. Possible FSM compositions are described by semantic composition rules. More complex structures can be obtained by introducing new FSMs and composition rules. This research will continue with the investigation of new validation and error correction procedures for semantic interpretation.

References

Bangalore, S., Johnston, M., 2004. Balancing data-driven and rule-based approaches in the context of a multimodal conversational system. In: Proc. HLT-NAACL Conf. Boston, Mass, USA, pp. 33–40.

Béchet, F., Nasr, A., Genet, F., 2000. Tagging unknown proper names using decision trees. In: 38th Annual Meeting Assoc. Comput. Linguist., Hong-Kong, China, pp. 77–84.

Béchet, F., Gorin, A. L., Wright, J. H., Hakkani-Tür, D., 2002. Named entity extraction from spontaneous speech in How May I Help You? In: Proc. ICSLP'02, Denver, Colorado.

Brachman, R., Schmolze, J., 1985. An overview of the KL-ONE knowledge representation system. Cognitive Science 9 (2), 171–216.

Chappelier, J., Rajman, M., Aragues, R., Rozenknop, A., 1999. Lattice parsing for speech recognition. In: Proc. 6th Conf. on Traitement Automat. du Langage Naturel TALN'99, Cargese, Corsica, France.

Estève, Y., Raymond, C., De Mori, R., Janiszek, D., 2003. On the use of linguistic consistency in systems for human–computer dialogs. IEEE Trans. Speech Audio Process. 11 (6), 746–756.

Hacioglu, K., 2004. A lightweight semantic chunker based on tagging. In: Proc. HLT-NAACL Conf. Boston, Mass, USA, pp. 145–148.

Hacioglu, K., Ward, W., 2001. A word graph interface for a flexible concept based speech understanding framework. In: Proc. Eurospeech-2001. Denmark.

Haffner, P., Tur, G., Wright, J., 2003. Optimizing SVMs for complex call classification. In: IEEE Internat. Conf. on Acoust. Speech Signal Process., ICASSP'03. Hong-Kong.

He, Y., Young, S., 2003. A data-driven spoken language understanding system. In: Automat. Speech Recognit. Understanding Workshop—ASRU'03. St. Thomas, US-Virgin Islands.

He, Y., Young, S., 2004. Robustness issues in a data-driven spoken language understanding system. In: Proc. Spoken Language Understanding for Conversational Syst. Higher Level Linguistic Inform. Speech Process. Boston, MA, pp. 39–46.

Jackendoff, R., 1990. Semantic structures. The MIT Press, Cambridge Mass.

Kaiser, E., Johnston, M., Heeman, P., 1999. PROFER: predictive, robust finite-state parsing for spoken language. In: Proc. IEEE Internat. Conf. Acoust., Speech Signal Process., ICASSP'99, Vol. 2. Phoenix, AZ.

Kuhn, R., De Mori, R., 1995. The application of semantic classification trees to natural language understanding. IEEE Trans. Pattern Anal. Mach. Intell. 17, 449–460.

Levesque, H.J., Brachman, R., 1985. A fundamental trade-off in knowledge Representation and Reasoning. In: Readings in Knowledge Representation. Morgan Kaufmann, pp. 42–70.

Levin, E., Pieraccini, R., 1995. Concept-based spontaneous speech understanding system. In: Proc. Eurospeech-95. Madrid, Spain, pp. 555–558.

Mohri, M., Pereira, F., Riley, M., 1997. AT&T FSM Library—Finite State Machine Library. AT&T Labs—Research. Available from: <http://www.research.att.com/sw/tools/fsm/>.

Mohri, M., Pereira, F., Riley, M., 2002. Weighted finite-state transducers in speech recognition. Comput. Speech Language 16 (1), 69–88.

Potamianos, A., Kuo, H., 2000. Statistical recursive finite state machine parsing for speech understanding. In: Proc. ICSLP-2000. Vol. 3. Beijing, China.

Pradhan, S., Ward, W., Hacioglu, K., Martin, J. H., Jurafsky, D., 2004. Shallow semantic parsing using support vector machines. In: Proc. HLT-NAACL Conf. Boston, MA, USA, pp. 33–40.

Rahim, M., Riccardi, G., Saul, L., Wright, J., H, Buntschuh, B., Gorin, A., L, 2001. Robust numeric recognition in spoken language dialogue. Speech Commun. 34, 195–212.

Raymond, C., Estève, Y., Béchet, F., Mori, R. D., Damnati, G., 2003. Belief confirmation in spoken dialogue systems using confidence measures. In: Automat. Speech Recognit. Understanding Workshop—ASRU'03. St. Thomas, US-Virgin Islands.

Roark, B., 2002. Markov parsing: lattice rescoring with a statistical parser. In: Proc. 40th ACL Meeting, Philadelphia.

Sadek, D., Ferrieux, A., Cozannet, A., Bretier, P., Panaget, F., Simonin, J., 1996. Effective human–computer cooperative spoken dialogue: the AGS demonstrator. In: ICSLP'96, USA.

Sarikaya, R., Gao, Y., Picheny, M., 2004. A comparison of rule–based and statistical methods for semantic language modeling and confidence measurement. In: Proc. HLT-NAACL Conf. Boston, USA, pp. 65–68.

Seneff, S., 1992. Tina: a natural language system for spoken language applications. Comput. Linguist. 18 (1), 61–86.

Vidal, E., Pieraccini, R., Levin, E., 1993. Learning associations between grammars: a new approach to natural language understanding. In: Proc. Eurospeech 93, Berlin, Germany.

Wang, Y.-Y., Acero, A., Chelba, C., Frey, B., Wong, L., 2002. Combination of statistical and rule-based approaches for spoken language understanding. In: Proc. Internat. Conf. Spoken Language Process.—ICSLP, Denver, CO.

Young, S.R., Hauptmann, A.G., Ward, W.H., Smith, E.T., Werner, P., 1989. High level knowledge sources in usable speech recognition systems. Commun. ACM 32 (2), 183–194.