Combining Acoustic Confidence Scores with Deep Semantic Analysis for Clarification Dialogues

Malte Gabsdil
Dept. of Comp. Linguistics
Saarland University
gabsdil@coli.uni-sb.de

Johan Bos
ICCS, Division of Informatics
The University of Edinburgh
jbos@cogsci.ed.ac.uk

Abstract

This paper describes a technique to include acoustic confidence scores as returned by automated speech recognisers in generic semantic representations. The method we propose requires only minimal changes to an existing grammar used for speech applications. Special attention is paid to the treatment of multi-word lexemes and combining several (N-best) speech recognition results into one semantic representation. The approach has been implemented and tested using the Nuance speech recognition software and a chart parser, in the formalism of underspecified discourse representations. The potential relevance of confidence scores in rich semantic representations is illustrated by generating more flexible clarification questions in dialogue systems.

1 Introduction

Many current commercial and research prototype spoken dialogue systems use *slot-filling* as a technique to recover semantic information from the speech recognition component. The speech recogniser thereby often returns slot-value pairs together with a *confidence score* which is calculated on the basis of the individual acoustic confidences of the words that trigger the value of a slot. The slot-value pairs, and their respective confidence scores, can then be used in subsequent processing by the dialogue manager to decide how the dialogue should proceed (e.g. whether clarification or confirmation of the user's utterance is needed or not). A case in point is the commercial Nuance speech recognition and dialogue management software (`http://www.nuance.com`) or Scansoft's ASR1600 SDK (former Lernout and Hauspie, `http://www.scansoft.com`).

The slot-filling paradigm is successful in several applications such as travel service and booking, reservation confirmation, price and availability information, and call routing. For instance, the Nuance system is able to return the following result from the recogniser for the utterance "Withdraw fifteen hundred dollars from savings":

Slot	**Value**	**Confidence**
action	withdraw	49
amount	$1500.00	72
account	savings	61

From this output, the dialogue manager might conclude that the user wants to perform some transaction on her savings account, but on the other hand decide that it has to confirm the requested type of transaction. Furthermore, the dialogue manager can make use of a set of slot-value pairs derived from the N-best recognition results to decide whether it should initiate an alternative clarification question (e.g. in situations where different values are assigned to the same slot in different hypotheses).

For speech applications that demand a deeper analysis of semantic information, the limitations of slot-filling approaches soon become an obstacle for serious semantic interpretation. Dialogue systems in the area of home automation show the need for a proper treatment of quantification and negation [7]. Speech-controlled mobile robots often have such a rich scenario of primitive actions that slot-filling is simply infeasible [6]. These systems employ richer semantic representations and apply ambiguity resolution in a second stage for sufficient understanding of speaker's utterances. In other research oriented approaches, such as collaborative problem solving [1] or tutorial dialogues, the need for a fine-grained semantics is even more obvious.

The aim of this paper is to enrich fine-grained semantic representations used in spoken dialogue systems with recogniser confidence scores. In particular, we will show how we can extend an existing grammar in a modular way employing only minimal changes in both the syntactic analysis and the corresponding semantic operations. We first consider single recognition results and then extend the approach to N-best processing. The result is a system with new potential for computational semantics, showing the merit of deep semantic analysis for generating clarification speech acts in spoken dialogue systems, using an existing commercial speech recogniser.

2 Confidence Scores and Clarification Dialogues

The main motivation for integrating acoustic confidence scores with deep semantic representations is to improve the quality of interaction in spoken dialogue systems, in particular with respect to clarification sub-dialogues.

The easiest way to decide whether clarification should be initiated for a user utterance is to look at its overall confidence score computed by the recogniser. This allows a system to ask the user to repeat/rephrase their input or to use implicit confirmation of the utterance in the next turn. However, it is often the case that some parts of an utterance are understood quite well and clarification is only needed for certain sub-phrases of the input. We already pointed out in the introduction that slot-based systems can exploit (sets of) slot confidences returned by the recogniser to generate clarification questions.

When we are dealing with richer semantic representations, however, the recogniser does not provide us with a mapping from semantic entities to acoustic confidences. We have to find a new way to integrate acoustic confidence scores with sub-formulas in the semantic representation and it is a much harder task to decide (1) which part of the input needs clarification, and (2) how to phrase the actual clarification question. Note that without considering individual word confidences these two questions cannot be decided at all (we could then only talk about the recognition result as a whole). Let us illustrate this with an example from the IBL corpus [6] using the Nuance speech recogniser. The recognition engine returns the best hypothesis for an input utterance together with an overall confidence score and individual confidence scores for each word. In the example below, the first number in the recognised string reflects the overall confidence score and the scores following each word in paranthesis indicate their individual confidence scores:

```
File 1:  u1_GA_MD_1.wav
Transcription:  again you walk straight ahead
Recognised(50):  again(66) you(60) walk(49) straight(28) ahead(58)
```

If we only consider the recognition result and its overall confidence score, we can only ask for clarification with respect to the whole utterance (e.g. "Do you want me to go straight ahead?"). We cannot focus on specific parts of the input utterance with particularly low confidence scores simply because this information is not available. Suppose now that we are also given the confidence scores on the word level but without a possibility to include them in the semantic representation. The best a system could do is asking "Did you say *straight*?" or simply "Straight?". In contrast, if we had a semantic representation augmented with confidence scores, we could interface to a domain ontology or the selectional restrictions of the verb to produce more informative clarifications like "Sorry, which direction?" or "Where shall I go?". Attaching confidence scores to sub-formulas in the semantic representation would allow us in this case to focus on a particular sub-constituent of the utterance and to select the appropriate WH-word (or phrase) needed to refer to this constituent. Another advantage of putting confidence scores in the semantic representation is that it allows us to pose clarification questions that relate to the *task* that the user has to perform [9]. Suppose the recogniser hypothesised "Walk past the bridge" with a low confidence for the preposition. In this case, the system might react with a question like "You mean towards Tescos?" if indeed there is an appropriate landmark in the computed direction.

We can also use the confidence scores to decide whether the system should pose alternative questions when faced with competing hypotheses (N-best processing). First of all, we can compare the scores for the alternatives and may find out that one of them is sufficiently better than the others. In this case there is no need to clarify. If, on the other hand, two alternatives have approximately the same confidence, we can check whether they make a difference for the task at all (e.g. small words like determiners often do not make a difference when the semantic representation is translated to action primitives). Only if there is a difference on the task level, we have to clarify and may use task-level reformulations as discussed above to make the differences in interpretations more apparent to the user. Finally, we imagine that confidence scores can also help us to pose clarification questions related to pronoun resolution. For example, if we have a high confidence score for a pronoun but cannot find an antecedent for it in the semantic representation, the system can come up with a question like "Who/what do you mean by he/it?".

3 Grammar Extension with Confidence Scores

Our approach to integrating confidence scores in semantic representations involves altering the grammar used for interpretation in such a way that it treats confidence scores as numbers that are part of the string to be parsed. This requires only minimal adaptations to the grammar rules, the corresponding semantic rules and the lexicon.

The semantic representations in our system are produced with the help of a unification grammar, using operations borrowed from the lambda-calculus as a "glue language", such as functional application and β-conversion. We refer to this as the *base grammar.* The base grammar is used to derive two grammars: a speech recognition grammar (in the Nuance GSL format, using the UNIANCE compiler [3]), and a confidence assignment grammar (see Fig. 1). The confidence assignment grammar is built by extending the base grammar in two ways: (1) extending the rule set to deal with confidence scores, and (2) integrating confidence scores into the lexical semantics. Note that, because the speech recognition grammar is a proper subset of the base grammar, and the confidence assignment grammar is only extended with respect to the coverage of confidence scores, it is guaranteed that any output of the speech recogniser is covered by the confidence assignment grammar.

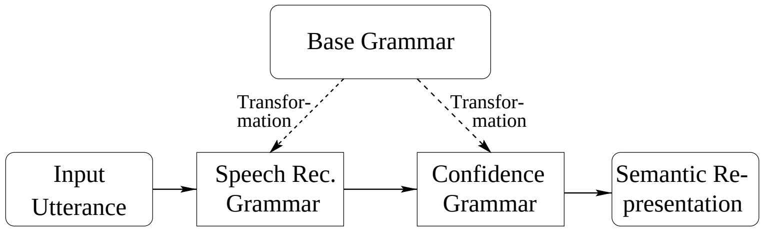


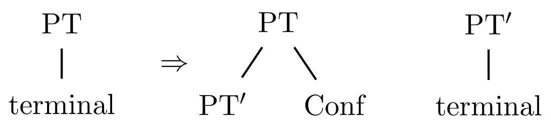
Figure 1: Deriving different grammars from one common base grammar.

It is common practice in dialogue system design to use a shared grammar resource which is then compiled out into different formats needed by the various components of the actual application. An example is SRI's GEMINI platform [5] which also generates Nuance GSL format and additionally includes a generation component.

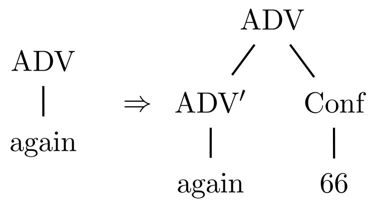
So what are the minimal changes to our grammar required to parse utterances including word confidence scores? First of all, we need to extend it with pre-terminal rules expanding to all possible confidence scores. As confidences are simply numerical values in the range $[0 \ldots 100]$, this will amount to the following rules, introducing the new syntactic category *Conf*:

$$\begin{array}{cccc} \text{Conf} & \text{Conf} & & \text{Conf} \\ | & | & \cdots & | \\ 0 & 1 & & 100 \end{array}$$

In a second step we have to replace each pre-terminal rule $PT \rightarrow \textit{terminal}$ with two new rules: one that combines with a confidence score, and one that replaces the original pre-terminal rule. This transformation can be depicted as follows:

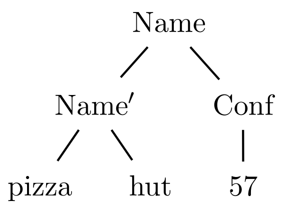


The modified grammar now allows us to parse utterances where each word is followed by a confidence score. For example, the (lexical) parse tree for the word 'again' now looks like this:



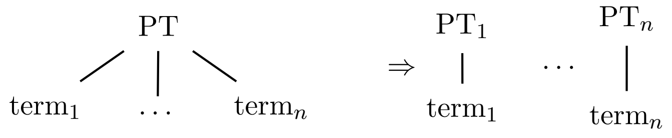
These transformational rules minimise the changes to our grammar because we only have to rewrite pre-terminal rules, which can easily be automated. The rest of the grammar is not affected. Another advantage of this method is that later we will only have to devise one additional semantic construction rule, namely the rule that combines our new pre-terminals with the confidence scores.[1]

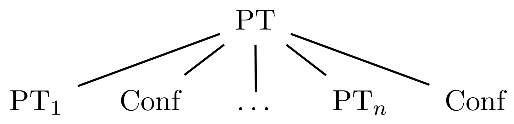
One obstacle in the transformation we need to deal with is a proper treatment of multi-word terminals generated by a single preterminal symbol. The problem is that all words within a multi-word sequence will be assigned a confidence score by the recogniser, but a naive extension of the transformation procedure only allows for a single confidence score at the end of each sequence:



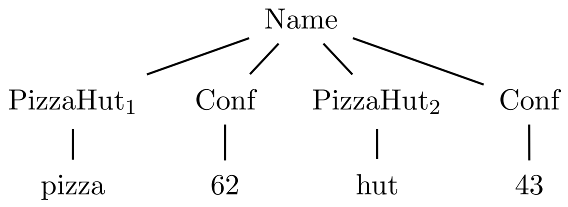
To remedy this problem, we choose to generalise the grammar transformation in a different way. We replace pre-terminal rules for multi-word lexemes by the following set of rules:

[1]We should note, however, that the transformation as it stands strictly requires that there is a single pre-terminal rule for each terminal symbol. That is, it cannot handle rules of the form $X \rightarrow \textit{terminal}\ Y$, where Y is a non-terminal. Rules of this form have to be substituted by two new rules: one where *terminal* is replaced by a unique pre-terminal P and a new pre-terminal rule $P \rightarrow \textit{terminal}$. We did not extend our transformation algorithm to deal with such cases because in our implementation there is a strict separation between lexical rules (pre-terminal rules) and all other grammar rules.





This allows for a confidence score after each word in the input utterance as produced by the speech recogniser. Given these new rules we now get the following parse tree for “pizza 62 hut 43”:



Another reason to choose such a flat analysis for multi-word sequences is to facilitate the semantic construction process in a smooth and straightforward way. As argued above for single word terminals, we only have to devise one additional semantic rule that, for multi-word terminals, combines several words and their confidences scores into one semantic representation for the mother category.

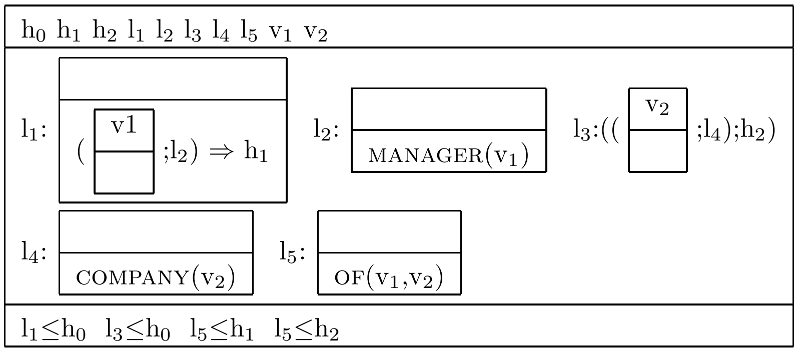
4 Semantic Representations and Confidence Scores

Now that we know how to parse a string decorated with confidence scores, we introduce the process of building semantic representations in the transformed grammar. Again, the changes that are required are minimal: we need to formulate the semantic rules mirroring the newly introduced grammar rules, and modify the lexical semantics of each category to incorporate the confidence scores.

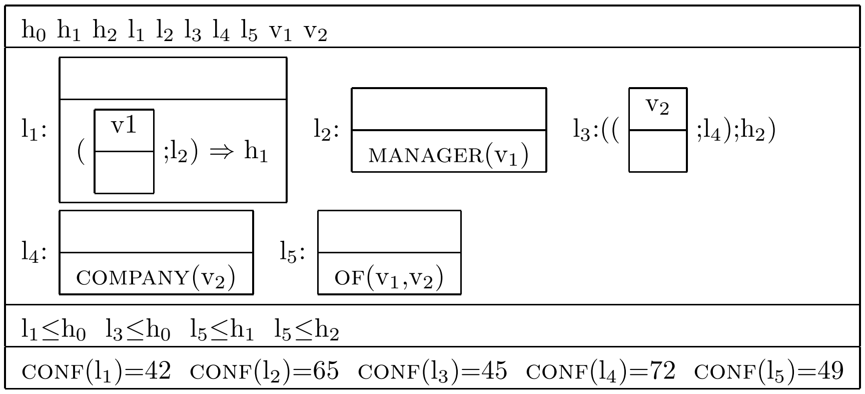
The semantic representations that we use are *underspecified discourse representations* (UDRs), adopted from Hole Semantics [2] and Reyle's underspecified DRSs [8]. The structures used in Hole Semantics represent ambiguities as a set of sub-structures with *holes*, governed by a set of constraints on how these holes can be filled. Here we take these sub-structures to be DRSs, and use labels to refer to them in the constraints. Resolving an underspecified representation amounts to plugging the holes with DRSs in such a way that no constraints are violated. The syntax of UDRs is defined as follows:

1. If U is a finite set of variables, and L is a finite set of UDR-conditions, and C is a finite set of scoping constraints, then the tuple $\langle U, L, C\rangle$ is a UDR;
2. If U_1 and U_2 are UDRs, then so is $(U_1;U_2)$;
3. If l is a label, $\mathcal{B}$ a proto-DRS, then $l{:}\mathcal{B}$ is a UDR-condition;
4. If U_1 and U_2 are UDRs, then $U_1 \bigvee U_2$ is a UDR-condition;
5. If l is a label and h is a hole, then $l \leq h$ and $l = h$ are UDR-constraints.

Basic UDRs are defined by Clause 1. The variables (ranging over holes and labels) introduced in the domain of a UDR bind occurrences of variables in the UDR-conditions and UDR-constraints. Clause 2 introduces merging of UDRs. The merge **;** for UDRs is dynamic (like the merge for DRSs), and for a UDR $(U_1;U_2)$, variables introduced in U_1 bind free occurrences in U_2. Clause 3 defines labelled DRSs as in the original formulation of hole semantics. Clause 4, disjunction, is used to encode lexical or structural (non scope) ambiguities. The scoping constraints (Clause 5) are relations over labels and holes and express dominance within tree structures. For instance, $l \leq h$ states that h dominates l, or in other words, label l is in the scope of hole h, or h out-scopes l. The equality constraint identifies a hole with a label. The proto-DRSs required by Clause 3 are like ordinary DRSs, but with the difference that labels or holes can occur in argument positions for DRSs. Consider an example UDR for "every manager of a company":

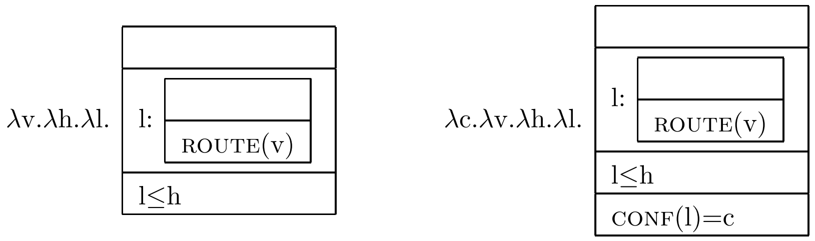


UDRs are used for semantic underspecification, representing ambiguities invoked by scope and anaphora in a compact way. In the example above, the nuclear scopes of ‘every manager’ and ‘a company’ are underspecified by the holes h_1 and h_2, respectively. To incorporate confidence scores, we will add a new modality to UDRs, and extend UDRs to tuples $\langle U, L, C, S\rangle$, where S is a finite set of values of confidence scores for labels in U. The use of labelling in multi-modal linguistic formalisms is not novel in itself—a case in point is the VIT formalism used in the Verbmobil project [10].


h_0 h_1 h_2 l_1 l_2 l_3 l_4 l_5 v_1 v_2
l_1: (v1 ;l_2) ⇒ h_1
l_2: MANAGER(v_1)
l_3:((v_2 ;l_4);h_2)
l_4: COMPANY(v_2)
l_5: OF(v_1,v_2)
$l_1 \leq h_0$ $l_3 \leq h_0$ $l_5 \leq h_1$ $l_5 \leq h_2$
CONF(l_1)=42 CONF(l_2)=65 CONF(l_3)=45 CONF(l_4)=72 CONF(l_5)=49

These are the semantic representations we work with. In order to integrate the recogniser’s confidence scores into UDRs, we have to address to further issues: (1) how to connect confidence scores for individual words to the appropriate sub-formulas in the semantic representation and (2) how to combine the lexical semantics of a word with a confidence score in a compositional way.

To attach a confidence score to a sub-formula we make use of the fact that each sub-formula in UDR is identified with a unique label and associate this label with the word’s confidence score. In order to do this, we have to identify the appropriate label for each word’s lexical semantic value (or *semantic macro* as we call it). Given this label, we can add a new condition CONF(l)=c, where c is the confidence score associated with the word label l. As we use the lambda calculus to steer the process of building semantic representations, we can add a further abstraction to lexical entries that deal with the confidence scores. For example, the original UDR and the modified one for the noun “route” look as follows:



Basically, the trick is to add a new lambda operator abstracting over the confidence score to each lexical semantic representation and use this as a functor which is applied to the semantic representation of the confidence score (in our case a number in the range $[0 \dots 100]$). The relevant semantic rule for the newly introduced grammar rule then looks as follows:

$$\text{PT:}\phi(\psi) \rightarrow \text{PT}'\text{:}\phi \ \text{Conf:}\psi$$

Let us consider a simple example. The semantic representation for "route 66", i.e., the word "route" recognised with a confidence score of 66, can now be compositionally constructed using function application and β-conversion.

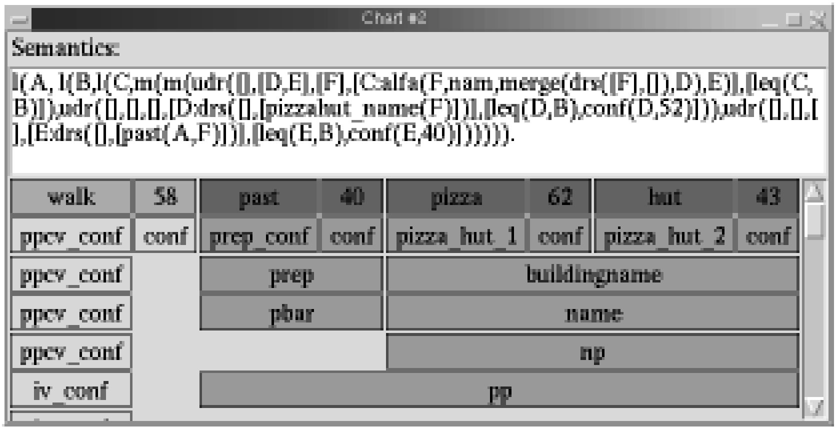
$$\lambda \text{c}.\lambda \text{v}.\lambda \text{h}.\lambda \text{l}.\begin{array}{|l|}\hline \\ \hline \text{l: } \begin{array}{|c|}\hline \\ \hline \textsc{route}(\text{v}) \\ \hline \end{array} \\ \hline \text{l}\leq\text{h} \\ \hline \textsc{conf}(\text{l})=\text{c} \\ \hline \end{array} (66) = \lambda \text{v}.\lambda \text{h}.\lambda \text{l}.\begin{array}{|l|}\hline \\ \hline \text{l: } \begin{array}{|c|}\hline \\ \hline \textsc{route}(\text{v}) \\ \hline \end{array} \\ \hline \text{l}\leq\text{h} \\ \hline \textsc{conf}(\text{l})=66 \\ \hline \end{array}$$

Finally, we turn to the issue of multi-word lexemes. In the lexicon they are treated as single words but with the new syntactic rules we introduced above we now have to combine several words and several confidence scores into one semantic representation. We do this by attaching the semantic representation of the entire former sequence to the first new terminal symbol introduced by our grammar transformation rule. The remaining words of the sequence will have an empty semantic value. As for the confidence scores, we simply calculate the unbiased average of the individual word confidences for the whole sequence. The semantic combination rule for multi-word sequences thus looks like follows:

$$\text{PT:}\phi\left(\frac{\sum_{i=1}^{n}\psi_i}{n}\right) \rightarrow \text{PT}_1\text{:}\phi \ \text{Conf:}\psi_1 \ \dots \ \text{PT}_n : \emptyset \ \text{Conf:}\psi_n$$

Preliminary experiments with the Nuance recogniser indicate that it might be better to use a weighted average based on the length of the individual words. The best results we achieved so far were obtained by looking up each word in the CMU pronunciation dictionary (`http://www.speech.cs.cmu.edu/cgi-bin/cmudict`) and weighting them by the number of phonemes (counting long vowels twice).

We implemented and tested the approach using a medium sized grammar applied to a corpus of spoken instructions to a mobile robot [6]. We used the Nuance 7.0 system for speech recognition, with a GSL grammar compiled from the base grammar using the UNIANCE compiler [3]. We used a bottom-up right-to-left chart parser to integrate the confidence scores in the semantic representations.



```
l(A, l(B,l(C,m(m(udr([],[D,E],[F],[C:alfa(F,nam,merge(drs([F],[]),D),E)],[leq(C,
B)]),udr([],[],[],[D:drs([],[pizzahut_name(F)])],[leq(D,B),conf(D,52)])),udr([],[],[
],[E:drs([],[past(A,F)])],[leq(E,B),conf(E,40)])))))).
```

Figure 2: Parser output for ‘walk 58 past 40 pizza 62 hut 43’

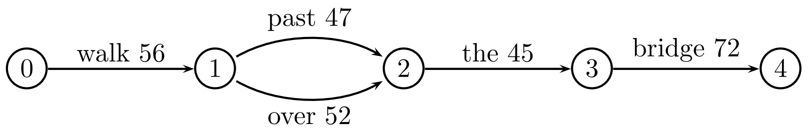
As an example for the semantic construction of a multi-word sequence we include a screen shot (Fig. 2) of our chart parser GUI for the sentence “walk 58 past 40 pizza 62 hut 43”. The upper part of Fig. 2 shows the semantics associated with the PP “past 40 pizza 62 hut 43”. The DRS label D is associated with the multi-word expression ‘pizza 62 hut 43’ and is assigned a confidence score of 52 (we use integer division to calculate the average). The preposition “past” is represented by the condition labelled with E and is assigned a confidence score of 40.

5 Parsing Word Lattices

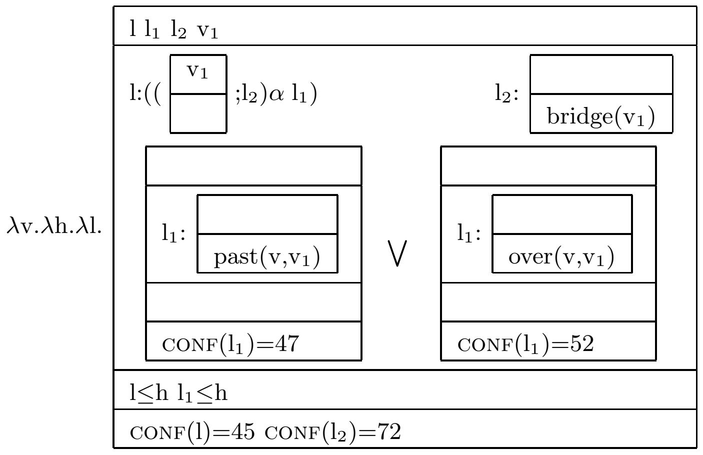
So far we have only discussed our approach with respect to the best result returned by the speech recogniser. Things get more complicated when we want to combine several analyses of the speech recogniser into one semantic representation. We are currently investigating the use of meta-level disjunction to represent different recogniser hypotheses in a compact way.

Here we consider the output of the recogniser structured as a word lattice. A word lattice is a topologically ordered directed acyclic graph representing all possible recogniser hypotheses. N-Best lists include the highest scoring paths through the lattice. We will illustrate this with an example.

Assume that the recogniser returned a word lattice instead of a single string of words (the Nuance 7.0 recogniser does not return lattices, but a set of N-best hypotheses). We can then define possible alternatives as partial paths that cover the same span in the lattice and are assigned the same syntactic category by the grammar. Consider the following simple made-up word lattice with confidence scores:



There are two hypotheses for the preposition in the lattice. One gives rise to the PP "past the bridge" and the other to the PP "over the bridge". In UDR, we can represent this by using meta-level disjunction.



To construct these representations we have to make two substantial changes to the parser. It has to (1) accept lattices (directed graphs) as input, and (2) combine the semantics of edges with the same syntactic category and the same start and end vertices during parsing.

The first change is rather simple: we represent lattices as a set of edges and initialise the agenda of the chart parser with all edges of the input lattice (see e.g. [11, 4]). The second modification is more complex and requires a change to the main parsing algorithm. Whenever we introduce a new edge into the chart we check whether an edge of the same syntactic category and same start and end vertices is already present in the chart. If not, we simply add the edge to the chart. Otherwise, we delete the existing edge from the chart and recursively all other edges that have it as a daughter. We then combine the semantics of this edge with the semantics of the new edge, add the resulting edge to the agenda, and resume processing.[2]

As for the combination of two semantic representations, we require that two edges of the same syntactic category also have the same semantic type. Two semantic representations can then be combined by unifying their lambda prefixes and joining their bodies by a meta-disjunctive operator. A schematic representation of the combination is given below, where $c_i = (a_i\ \mathit{unif}\ b_i)$.

$$\left\{\begin{array}{c}\lambda a_1.\lambda a_2.\ \dots\ \lambda a_m.U_a\\ \lambda b_1.\lambda b_2.\ \dots\ \lambda b_m.U_b\end{array}\right\} \Rightarrow \lambda c_1.\lambda c_2.\ \dots\ \lambda c_m.\begin{array}{|c|}\hline \quad \\ \hline \mathrm{U}_a \vee \mathrm{U}_b \\ \hline \quad \\ \hline \quad \\ \hline\end{array}$$

Note that by unifying the abstracted variables we do not introduce any free variables. All variables that where bound before combination are also bound after combination. Further note that in-situ β-conversion of the semantic representation is required each time a new edge is produced in order to guarantee combination of two categories using the above rule.

6 Conclusions

We presented a method to include acoustic confidence scores as returned by automated speech recognisers in generic semantic representations, requiring only minimal changes to an existing grammar used for speech applications. We are not aware of any previous work aiming to reach this goal. We implemented our formalism using off-the-shelf commercial speech recognition software.

A promising application area for this kind of multi-modal semantic representation is the generation of clarification questions in spoken dialogue systems, and we are currently investigating ways to implement such a utilisation. One idea is to combine confidence scores for sub-formulas recursively and decide which constituent we want to clarify.

Of course, speech recognition confidence scores are not the only sources that might trigger clarification dialogues. Other factors include ambiguities (attachment, scope, reference) as well as plausibility, relevance, consistency and informativeness of the utterance in a specific situation. Future research

[2]We are aware that this approach results in re-parsing the same syntactic structures with new semantic values. A more efficient way of doing this would be to replace the semantics of all relevant edges in the chart by the new combined semantics.

should address the issue of combining these elements for generating clarification questions in spoken dialogue systems.

References

[1] J. F. Allen, L. K. Schubert, G. M. Ferguson, P. A. Heeman, C. H. Hwang, T. Kato, M. N. Light, N. G. Martin, B. W. Miller, M. Poesio, and D. R. Traum. The TRAINS project: A case study in building a conversational planning agent. *Journal of Experimental and Theoretical AI*, 7:7–48, 1995.

[2] Johan Bos. Predicate Logic Unplugged. In P. Dekker and M. Stokhof, editors, *Proceedings of the Tenth Amsterdam Colloquium*, pages 133–143, ILLC/Dept. of Philosophy, University of Amsterdam, 1996.

[3] Johan Bos. Compilation of unification grammars with compositional semantics to speech recognition packages. In *COLING 2002. Proceedings of the 19th International Conference on Computational Linguistics*, pages 106–112, Taipei, Taiwan, 2002.

[4] J.-C. Chappelier, M. Rajman, R. Aragues, and A. Rozenknop. Lattice parsing for speech recognition. In *6ème conférence sur le Traitement Automatique du Langage Naturel (TALN99)*, Cagèse, France, July 1999.

[5] J. Dowding, J. Gawron, D. Appelt, J. Bear, L. Cherny, R. Moore, and D. Moran. GEMINI: A natural language system for spoken-language understanding. In *ACL-93*, pages 54–61, 1993.

[6] Stanislao Lauria, Guido Bugmann, Theocharis Kyriacou, Johan Bos, and Ewan Klein. Training Personal Robots Using Natural Language Instruction. *IEEE Intelligent Systems*, pages 38–45, Sept./Oct. 2001.

[7] Jose F. Quesada, Federico Garcia, Ester Sena, Jose Angel Bernal, and Gabriel Amores. Dialogue management in a home machine environment: Linguistic components over an agent architecture. In *Proceedings of the 17th SEPLN*, pages 89–98, 2001.

[8] Uwe Reyle. Dealing with Ambiguities by Underspecification: Construction, Representation and Deduction. *Journal of Semantics*, 10:123–179, 1993.

[9] Carolyn Penstein Rosé. *Robust Interactive Dialogue Interpretation.* PhD thesis, School of Computer Science, Carnegie Mellon University, 1997.

[10] Michael Schiehlen, Johan Bos, and Michael Dorna. Verbmobil interface terms (vits). In Wolfgang Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation.* Springer, 2000.

[11] Gertjan van Noord, Gosse Bouma, Rob Koeling, and Mark-Jan Nederhof. Robust grammatical analysis for spoken dialogue systems. *Natural Language Engineering*, 1(1):1–48, 1998.