Efficient Parsing of Spoken Inputs for Human-Robot Interaction

Pierre Lison and Geert-Jan M. Kruijff

***Abstract*— The use of deep parsers in spoken dialogue systems is usually subject to strong performance requirements. This is particularly the case in human-robot interaction, where the computing resources are limited and must be shared by many components in parallel. A real-time dialogue system must be capable of responding quickly to any given utterance, even in the presence of noisy, ambiguous or distorted input. The parser must therefore ensure that the number of analyses remains bounded at every processing step.**

The paper presents a practical approach to addressing this issue in the context of deep parsers designed for spoken dialogue. The approach is based on a word lattice parser combined with a statistical model for parse selection. Each word lattice is parsed incrementally, word by word, and a discriminative model is applied at each incremental step to prune the set of resulting partial analyses. The model incorporates a wide range of linguistic and contextual features and can be trained with a simple perceptron. The approach is fully implemented as part of a spoken dialogue system for human-robot interaction. Evaluation results on a Wizard-of-Oz test suite demonstrate significant improvements in parsing time.

I. Introduction

Most dialogue systems developed nowadays for human-robot interaction are based on crude processing methods such as keyword spotting or heuristic rules. These methods are undoubtedly useful for well-structured tasks definable by a set of slot-value pairs, but do not extend very well to more complex interactions, because they are insensitive to the syntactic and semantic structure of the utterance. To capture these linguistic relations, we need to build fine-grained *grammars* of natural language, as well as *parsers* operating on these grammars. Yet, the development of robust and efficient parsers for spoken dialogue is hindered by several major difficulties which need to be addressed.

The first difficulty is the pervasiveness of *speech recognition errors*. Automatic speech recognition is a highly error-prone task, and parsers designed to process spoken input must therefore be able to accomodate the various recognition errors that may arise. This problem is particularly acute for robots operating in real-world environments and dealing with utterances pertaining to complex, open-ended domains.

The second issue is the *relaxed grammaticality* of spoken language. Dialogue utterances are often incomplete, fragmentary or ungrammatical, and may contain numerous disfluencies like fillers (err, uh, mm), repetitions, self-corrections, etc. This is natural behaviour in human-human interaction [1] and can also be observed in several domain-specific corpora for HRI [2]. Spoken dialogue parsers should therefore be made robust to such ill-formed utterances.

Finally, the vast majority of spoken dialogue systems are designed to operate in *real-time*. This has two important consequences. First, the parser should not wait for the utterance to be completed to start processing it – instead, the set of possible semantic interpretations should be gradually built and extended as the utterance unfolds. Second, each processing step should operate under strict time constraints. The main obstacle here is the high level of ambiguity in natural language, which can lead to a combinatorial explosion in the number of possible readings.

The remainder of this paper is devoted to addressing this last issue, building on an integrated approach to situated spoken dialogue processing previously outlined in [3], [4]. The approach we present here is similar to [5], with some notable differences concerning the parser (our parser being specifically tailored for robust spoken dialogue processing), and the features included in the discriminative model.

An overview of the paper is as follows. We first describe in Section II the cognitive architecture in which our system has been integrated. We then discuss the approach in detail in Section III. Finally, we present in Section IV the quantitative evaluations on a WOZ test suite, and conclude.

II. Architecture

The approach we present in this paper is fully implemented and integrated into a cognitive architecture for autonomous robots. A recent description of the architecture is provided in [6], [7]. It is capable of building up visuo-spatial models of a dynamic local scene, and continuously plan and execute manipulation actions on objects within that scene. The robot can discuss objects and their material- and spatial properties for the purpose of visual learning and manipulation tasks. Figure 1 illustrates the architecture schema for the communication subsystem, limited to the comprehension side.

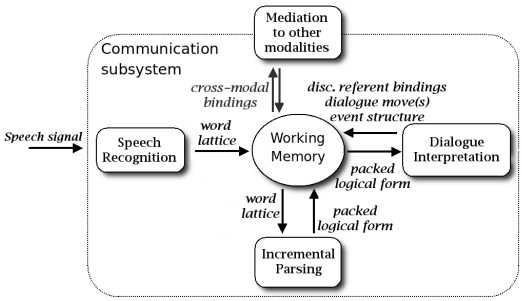

Fig. 1. Architecture schema of the communication subsystem.

This work was supported by the EU FP7 ICT Integrated Project "CogX" (FP7-ICT- 215181).

Pierre Lison and Geert-Jan M. Kruijff are with the German Research Centre for Artificial Intelligence (DFKI GmbH), Language Technology Lab, Saarbrücken, Germany {pierre.lison,gj} @ dfki.de

Starting with automatic speech recognition (ASR), we process the audio signal to establish a *word lattice* containing statistically ranked hypotheses about word sequences. Subsequently, parsing constructs grammatical analyses for the given (partial) word lattice. A grammatical analysis constructs both a syntactic analysis of the utterance, and a representation of its meaning. The analysis is based on an incremental chart parser[1] for Combinatory Categorial Grammar [8]. These meaning representations are ontologically richly sorted, relational structures, formulated in a (propositional) description logic – more precisely in the HLDS formalism [9]. An example of logical form expressed in HLDS is provided in the Figure 3. The parser itself is based on a variant of the CKY algorithm [10].

Once all the possible (partial) parses for a given (partial) utterance are computed, they are filtered in order to retain only the most likely interpretation(s). This ensures that the number of parses at each incremental step remains bounded and avoids a combinatorial explosion of the search space. The task of selecting the most likely parse(s) among a set of possible ones is called *parse selection*. We describe this task in detail in the next section.

At the level of dialogue interpretation, the logical forms are then resolved against a dialogue model to establish coreference and dialogue moves.

Linguistic interpretations must finally be associated with extra-linguistic knowledge about the environment – dialogue comprehension hence needs to connect with other subarchitectures like vision, spatial reasoning or planning. We realise this information binding between different modalities via a specific module, called the "binder", which is responsible for the ontology-based *mediation* accross modalities [11].

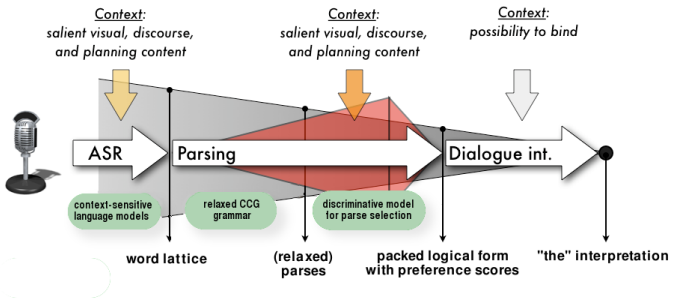

Fig. 2. Context-sensitivity in processing situated dialogue

Interpretation *in context* indeed plays a crucial role in the comprehension of utterance as it unfolds. Human listeners continuously integrate linguistic inputs with scene understanding (foregrounded entities and events) and world knowledge. These information sources are used both for interpreting what *has been* said, and for predicting/anticipating what is *going to be* said. Their integration is also closely *time-locked*, as evidenced by analyses of saccadic eye movements in visual scenes [12] and by neuroscience-based studies of event-related brain potentials [13].

[1]Built using the OpenCCG API: http://openccg.sf.net

Several approaches in situated dialogue for human-robot interaction demonstrated that a robot's understanding can be substantially improved by relating utterances to the situated context [14], [15], [7]. By incorporating contextual information at the core of our model, our approach also seeks to exploit this important insight.

III. Approach

As we just explained, the parse selection module is responsible for selecting at each incremental step a subset of "good" parses. Once the selection is made, the best analyses are kept in the parse chart, while the others are discarded and pruned from the chart.

A. The parse selection task

To achieve this selection, we need a mechanism to discriminate among the possible parses. This is done via a statistical model covering a large number of features.

Formally, the task is defined as a function $F : \mathscr{X} \rightarrow \mathscr{Y}$ where the domain $\mathscr{X}$ is the set of possible inputs (in our case, $\mathscr{X}$ is the set of possible *word lattices*), and $\mathscr{Y}$ the set of parses. We assume:

1) A function $\mathbf{GEN}(x)$ enumerating all possible parses for an input x. In our case, the function represents the admissible parses of the CCG grammar.
2) A d-dimensional feature vector $\mathbf{f}(x,y) \in \Re^d$, representing specific features of the pair (x,y). It can include various linguistic or contextual features.
3) A parameter vector $\mathbf{w} \in \Re^d$.

The function F, mapping a word lattice to its most likely parse, is then defined as:

$$F(x) = \underset{y \in \mathbf{GEN}(x)}{\operatorname{argmax}} \mathbf{w}^T \cdot \mathbf{f}(x,y) \qquad (1)$$

where $\mathbf{w}^T \cdot \mathbf{f}(x,y)$ is the inner product $\sum_{s=1}^{d} w_s \, f_s(x,y)$, and can be seen as a measure of the "quality" of the parse. Given the parameters $\mathbf{w}$, the optimal parse of a given utterance x can be therefore easily determined by enumerating all the parses generated by the grammar, extracting their features, computing the inner product $\mathbf{w}^T \cdot \mathbf{f}(x,y)$, and selecting the parse with the highest score.

The task of parse selection is an example of a *structured classification problem*, which is the problem of predicting an output y from an input x, where the output y has a rich internal structure. In the specific case of parse selection, x is a word lattice, and y a logical form.

B. Training data

In order to estimate the parameters $\mathbf{w}$, we need a set of training examples. Unfortunately, no corpus of situated dialogue adapted to our task domain is available to this day, let alone semantically annotated. The collection of in-domain data via Wizard of Oz experiments being a very costly and time-consuming process, we followed the approach advocated in [16] and *generated* a corpus from a hand-written task grammar.

Practically, we first collected a small set of WoZ data, totalling about a thousand utterances related to a simple scenario of object manipulation and visual learning. This set is too small to be directly used for statistical training, but sufficient to capture the most frequent linguistic constructions used in this particular context. Based on it, we designed a domain-specific grammar covering most of the utterances. Each rule is associated to a semantic HLDS representation. Weights are automatically assigned to each rule by parsing our corpus, leading to a small *stochastic context-free grammar* augmented with semantic information.

The grammar is then randomly traversed a large number of times, resulting in a set of around 25.000 utterances along with their semantic representations. To simulate the speech recognition errors, we synthesise each generated string using a text-to-speech engine[2], feed the audio stream to the speech recogniser, and retrieve the recognition result.

Via this technique, we are able to easily collect a large amount of training data. Because of its relatively artificial character, the quality of such training data is naturally lower than what could be obtained with a genuine corpus. But, as the experimental results will show, it remains sufficient to train the perceptron for the parse selection task, and achieve significant improvements in accuracy and robustness[3].

C. Perceptron learning

The algorithm we use to estimate the parameters $\mathbf{w}$ using the training data is a **perceptron**. The algorithm is fully online - it visits each example in turn, in an incremental fashion, and updates $\mathbf{w}$ if necessary. Albeit simple, the algorithm has proven to be very efficient and accurate for the task of parse selection [5], [17].

The pseudo-code for the online learning algorithm is detailed in [**Algorithm 1**].

It works as follows: the parameters $\mathbf{w}$ are first initialised to some arbitrary values. Then, for each pair (x_i, z_i) in the training set, the algorithm searchs for the parse y' with the highest score according to the current model. If this parse happens to match the best parse which generates z_i (which we shall denote y^*), we move to the next example. Else, we perform a simple perceptron update on the parameters:

$$\mathbf{w} = \mathbf{w} + \mathbf{f}(x_i, y^*) - \mathbf{f}(x_i, y') \quad (2)$$

The iteration on the training set is repeated T times, or until convergence.

Algorithm 1 Online perceptron learning

Require: - Set of n training examples $\{(x_i, z_i) : i = 1...n\}$
- For each incremental step j with $0 \leq j \leq |x_i|$, we define the partially parsed utterance x_i^j and its gold standard semantics z_i^j
- T: number of iterations over the training set
- GEN(x): function enumerating possible parses for an input x, according to the CCG grammar.
- GEN(x, z): function enumerating possible parses for an input x and which have semantics z, according to the CCG grammar.
- $L(y)$ maps a parse tree y to its logical form.
- Initial parameter vector $\mathbf{w_0}$

```
% Initialise
w ← w_0
% Loop T times on the training examples
for t = 1 ... T do
  for i = 1 ... n do
    % Loop on the incremental parsing steps
    for j = 0...|x_i| do
      % Compute best parse according to model
      Let y' = argmax_{y∈GEN(x_i^j)} w^T · f(x_i^j, y)
      % If the decoded parse ≠ expected parse, update the parameters of the model
      if L(y') ≠ z_i^j then
        % Search the best parse for the partial utterance x_i^j with semantics z_i^j
        Let y* = argmax_{y∈GEN(x_i^j, z_i^j)} w^T · f(x_i^j, y)
        % Update parameter vector w
        Set w = w + f(x_i^j, y*) − f(x_i^j, y')
      end if
    end for
  end for
end for
return parameter vector w
```

D. Features

In the previous section, we explained that the parse selection operates by enumerating the possible parses and selecting the one with the highest score according to the linear model parametrised by the weights $\mathbf{w}$. The accuracy of our method crucially relies on the selection of "good" features $\mathbf{f}(x, y)$ for our model - that is, features which help *discriminating* the parses. In our model, the features are of four types: semantic features, syntactic features, contextual features, and speech recognition features.

1) Semantic features: Semantic features are defined on *substructures* of the logical form (see Figure 3 for an example). We define features on the following information sources: the nominals, the ontological sorts of the nominals, the dependency relations (following [19]), and the sequences of dependency relations.

The features on nominals and ontological sorts aim at modeling (aspects of) *lexical semantics* - e.g. which meanings are the most frequent for a given word -, whereas the features on relations and sequence of relations focus on *sentential semantics* - which dependencies are the most frequent. These features therefore help us handle various forms of lexical and syntactic ambiguities.

[2]We used MARY (http://mary.dfki.de) for the text-to-speech engine.

[3]In a near future, we plan to progressively replace this generated training data by a real spoken dialogue corpus adapted to our task domain.

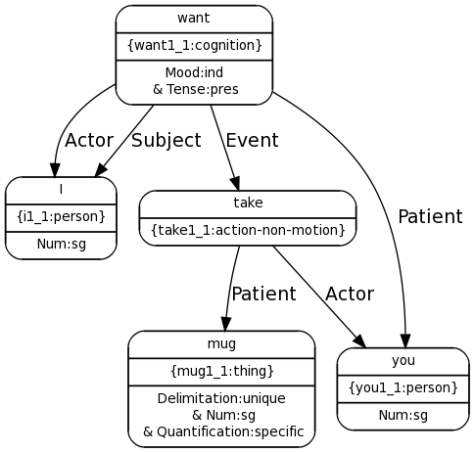

Fig. 3. HLDS logical form for "*I want you to take the mug*".

2) Syntactic features: Syntactic features are features associated to the *derivational history* of a specific parse. Alongside the usual CCG rules (application, composition and type raising), our parser also uses a set of non-standard rules designed to handle disfluencies, speech recognition errors, and fragmentary utterances by selectively relaxing the grammatical constraints (see [4] for details).

In order to "*penalise*" to a correct extent the application of these non-standard rules, we include in the feature vector $\mathbf{f}(x,y)$ new features counting the number of times these rules are applied in the parse. In the derivation shown in the Figure 4, the rule *corr* (correction of a speech recognition error) is for instance applied once.

$$\dfrac{\dfrac{\text{pick}}{\text{s/particle/np}}\quad\dfrac{\dfrac{\text{cup}}{\text{up}}\ \textit{corr}}{\text{particle}}}{\text{s/np}}\ >\quad\dfrac{\dfrac{\text{the}}{\text{np/n}}\quad\dfrac{\text{ball}}{\text{n}}}{\text{np}}\ >}{\text{s}}\ >$$

Fig. 4. CCG derivation of *"pick cup the ball"*.

These syntactic features can be seen as a *penalty* given to the parses using these non-standard rules, thereby giving a preference to the "normal" parses over them. This ensures that the grammar relaxation is only applied "as a last resort" when the usual grammatical analysis fails to provide a parse.

3) Contextual features: As we have already outlined in the background section, one striking characteristic of spoken dialogue is the importance of *context*. Understanding the visual and discourse contexts is crucial to resolve potential ambiguities and compute the most likely interpretation(s) of a given utterance.

The feature vector $\mathbf{f}(x,y)$ therefore includes various features related to the context:

- *Activated words*: our dialogue system maintains in its working memory a list of contextually activated words (cfr. [20]). This list is continuously updated as the dialogue and the environment evolves. For each context-dependent word, we include one feature counting the number of times it appears in the utterance string.
- *Expected dialogue moves*: for each possible dialogue move, we include one feature indicating if the dialogue move is consistent with the current discourse model. These features ensure for instance that the dialogue move following a QuestionYN is a Accept, Reject or another question (e.g. for clarification requests), but almost never an Opening.

4) Speech recognition features: Finally, the feature vector $\mathbf{f}(x,y)$ also includes features related to the *speech recognition*. The ASR module outputs a set of (partial) recognition hypotheses, packed in a word lattice. One example is given in Figure 5.

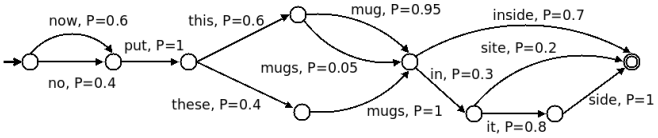

Fig. 5. Example of word lattice

We want to favour the hypotheses with high confidence scores, which are, according to the statistical models incorporated in the ASR, more likely to reflect what was uttered. To this end, we introduce in the feature vector several acoustic features measuring the likelihood of each recognition hypothesis.

E. Incremental chart pruning

In the previous subsections, we explained how the parse selection was performed, and on basis of which features. We now detail how this operation can be exploited to improve the parsing performance of the dialogue system.

The efficiency of a parser is crucially dependent on its treatment of *ambiguities*. Natural language – and especially spoken language – is indeed inherently ambiguous, at all linguistic levels (morphological, syntactic, semantic, pragmatic). If left unchallenged, this characteristic can lead to a combinatorial explosion in the number of possible readings, and, as a logical consequence, to a fast degradation of the parser performance. This fate can be prevented by pruning the generated analyses at each processing step, and keeping only a limited number of interpretations.

This is implemented in our system via the parse selection module, which is triggered at each incremental step and used to discriminate between the "good" parses that needs to be kept in the parse chart, and those that should be pruned.

To this end, we introduce a new parameter in our parser: the *beam width*. The beam width defines the maximal number of analyses which can be kept in the chart at each incremental step. If the number of possible readings exceeds the beam width, the analyses with a lower parse selection score are removed from the chart.

Practically, this is realised by removing the top signs associated in the chart with the set of analyses to prune,

	Beam width	Size of word lattice	Average parsing time per utterance (in s.)	Exact-match			Partial-match		
				Precision	Recall	F_1-value	Precision	Recall	F_1-value
(Baseline)	*(none)*	*10*	*10.1*	*40.4*	*100.0*	*57.5*	*81.4*	*100.0*	*89.8*
	120	10	5.78	40.9	96.9	57.5	81.9	98.0	89.2
	60	10	**4.82**	41.1	92.5	56.9	81.7	94.1	87.4
	40	10	4.66	39.9	88.1	54.9	79.6	91.9	85.3
	30	10	4.21	41.0	83.0	54.9	80.2	88.6	84.2
	20	10	4.30	40.1	80.3	53.5	78.9	86.5	82.5
(Baseline)	*(none)*	*5*	*5.28*	*40.0*	*100.0*	*57.1*	*81.5*	*100.0*	*89.8*
	120	5	6.62	40.9	98.4	57.8	81.6	98.5	89.3
	60	5	5.28	40.5	96.9	57.1	81.7	97.1	88.7
	40	5	4.26	40.9	91.0	56.5	81.7	92.4	86.7
	30	5	**3.51**	40.7	92.4	56.5	81.4	93.9	87.2
	20	5	2.81	36.7	87.1	51.7	79.6	90.7	84.8

TABLE I

EVALUATION RESULTS (IN SECONDS FOR THE PARSING TIME, IN % FOR THE EXACT- AND PARTIAL-MATCH).

as well as all the intermediate signs which are included in these top signs *and* are not used in any of the "good" analyses retained by the parse selection module.

A simple backtracking mechanism is also implemented in the parser. In case the beam width happens to be too narrow and renders the utterance unparsable, it is possible to reintroduce the signs previously removed from the chart and restart the parse at the failure point.

The combination of incremental parsing and incremental chart pruning provides two decisive advantages over classical, non-incremental parsing techniques: first, we can start processing the spoken inputs as soon as a partial analysis can be outputted by the speech recogniser. Second, the pruning mechanism ensures that each incremental parsing step remains time-bounded. Such a combination is therefore ideally suited for the real-time spoken dialogue systems used in human-robot interaction.

IV. EVALUATION

We performed a quantitative evaluation of our approach, using its implementation in a fully integrated system (cf. Section II). To set up the experiments for the evaluation, we have gathered a Wizard-of-Oz corpus of human-robot spoken dialogue for our task-domain (Figure 6), which we segmented and annotated manually with their expected semantic interpretation. The data set contains 195 individual utterances [4] along with their complete logical forms. The utterances were free both in form and in content, and could include any kind of commands, questions, assertions, clarifications, etc. related to the scenario. The utterance length in the collected WoZ corpus varies from 1 to 16 words, with an average length of 6.1 words.

The results are shown in the Table I. We tested our approach for five different values of the beam width parameter, and for two sizes of the word lattice (indicated by the horizontal division in the middle of the table). The results are compared against a baseline, which is the performance of our parser without chart pruning. For each configuration, we give the average parsing time, as well as the exact-match and partial-match results (in order to verify that the performance increase is not cancelled by a drop in accuracy).

We define the precision, recall, and F_1 in terms of true positives (TP), false positives (FP) and false negatives (FN):

$$\text{TP} = \#\text{utterances correctly matched} \tag{3}$$

$$\text{FP} = \#\text{utterances incorrectly matched} \tag{4}$$

$$\text{FN} = \#\text{utterances with no computed interpretation} \tag{5}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \qquad \text{recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{6}$$

$$F_1 = 2 \times \frac{(\text{precision} \times \text{recall})}{\text{precision} + \text{recall}} \tag{7}$$

The evaluation results demonstrate very significant improvements compared to the baseline, with notably a **52.2** % decrease in parsing time for a word lattice of size 10 and a beam width of 60.

It is also worth noting that the choice of the beam width parameter is crucial. Above 30-40, the pruning mechanism is very efficient – we observe a notable decrease in parsing time without any real accuracy degradation. Below this threshold, the beam width is too small to retain all the necessary information in the chart and the recall quickly drops.

Fig. 6. WoZ experiments for object manipulation and visual learning

Figure 7 illustrates the evolution of the ambiguity level (in terms of number of alternative semantic interpretations) during the incremental parsing. We observe that the chart pruning mechanism acts as a *stabilising factor* within the

[4] More precisely, word lattices provided by the speech recogniser. These word lattices can contain a maximum of 10 recognition hypotheses.

parser, by limiting the number of ambiguities produced after every incremental step to a reasonable level.

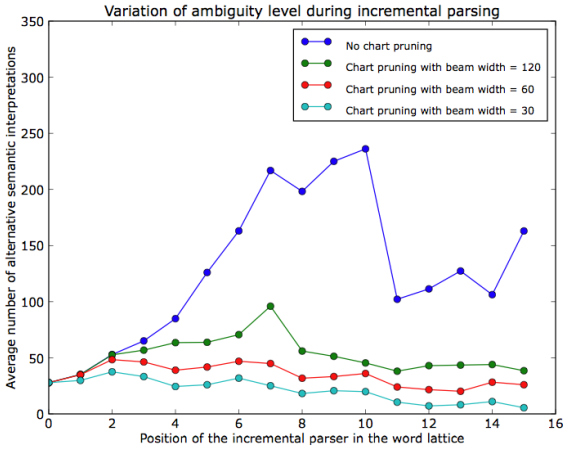

Fig. 7. Variation of ambiguity level during incremental parsing, with and without chart pruning (on word lattices with NBest 10 hypotheses).

V. Conclusions

We presented in this paper an original approach for efficient parsing of spoken inputs, based on a combination of incremental *parsing* – to start processing as soon as a partial speech input is recognised – and incremental *chart pruning* – to limit at each step the number of retained analyses.

The incremental parser is based on a fine-grained Combinatory Categorial Grammar, and takes ASR word lattices as input. It outputs a set of partial semantic interpretations ("logical forms"), which are progressively refined and extended as the utterance unfolds.

Once the partial interpretations are computed, they are subsequently pruned/filtered to retain only the most likely hypotheses in the chart. This mechanism is based on a *discriminative model* exploring a set of relevant semantic, syntactic, contextual and acoustic features extracted for each parse. The model yields a score for each resulting partial hypothesis, and the parser only maintains in its chart a limited number of high-scoring parses, and prune the others.

The experimental evaluation conducted on a Wizard-of-Oz test suite demonstrated that the aforementioned approach was able to significantly improve the parser performance .

As forthcoming work, we shall examine the extension of our approach in new directions, such as the introduction of more refined contextual features, the extension of the grammar relaxation rules, or the use of more sophisticated learning algorithms such as Support Vector Machines.

References

[1] R. Fernández and J. Ginzburg, "A corpus study of non-sentential utterances in dialogue," *Traitement Automatique des Langues*, vol. 43, no. 2, pp. 12–43, 2002.

[2] E. A. Topp, H. Hüttenrauch, H. Christensen, and K. Severinson Eklundh, "Bringing together human and robotic environment representations – a pilot study," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, October 2006.

[3] P. Lison, "Robust processing of situated spoken dialogue," Master's thesis, Universität des Saarlandes, Saarbrücken, 2008. [Online]. Available: http://www.dfki.de/~plison/pubs/thesis/main.thesis.plison2008.pdf

[4] P. Lison and G.-J. M. Kruijff, "An integrated approach to robust processing of situated spoken dialogue," in *Proceedings of the Second International Workshop on the Semantic Representation of Spoken Language (SRSL'09)*, Athens, Greece, 2009.

[5] M. Collins and B. Roark, "Incremental parsing with the perceptron algorithm," in *ACL '04: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2004, p. 111.

[6] N. A. Hawes, A. Sloman, J. Wyatt, M. Zillich, H. Jacobsson, G.-J. M. Kruijff, M. Brenner, G. Berginc, and D. Skocaj, "Towards an integrated robot with multiple cognitive functions." in *Proc. AAAI'07*. AAAI Press, 2007, pp. 1548–1553.

[7] G.-J. M. Kruijff, P. Lison, T. Benjamin, H. Jacobsson, and N. Hawes, "Incremental, multi-level processing for comprehending situated dialogue in human-robot interaction," in *Language and Robots: Proceedings from the Symposium (LangRo'2007)*, Aveiro, Portugal, December 2007, pp. 55–64.

[8] M. Steedman and J. Baldridge, "Combinatory categorial grammar," in *Nontransformational Syntax: A Guide to Current Models*, R. Borsley and K. Börjars, Eds. Oxford: Blackwell, 2009.

[9] J. Baldridge and G.-J. M. Kruijff, "Coupling CCG and hybrid logic dependency semantics," in *ACL'02: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, PA: Association for Computational Linguistics, 2002, pp. 319–326.

[10] T. Kasami, "An efficient recognition and syntax algorithm for context-free languages," Air Force Cambridge Research Laboratory, Bedford, Massachusetts, Tech. Rep. AF-CRL-65-758, 1965.

[11] H. Jacobsson, N. Hawes, G.-J. Kruijff, and J. Wyatt, "Crossmodal content binding in information-processing architectures," in *Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Amsterdam, The Netherlands, 2008.

[12] P. Knoeferle and M. Crocker, "The coordinated interplay of scene, utterance, and world knowledge: evidence from eye tracking," *Cognitive Science*, 2006.

[13] J. Van Berkum, "Sentence comprehension in a wider discourse: Can we use ERPs to keep track of things?" in *The on-line study of sentence comprehension: Eyetracking, ERPs and beyond*, M. Carreiras and C. C. Jr., Eds. New York NY: Psychology Press, 2004, pp. 229–270.

[14] D. Roy, "Semiotic schemas: A framework for grounding language in action and perception," *Artificial Intelligence*, vol. 167, no. 1-2, pp. 170–205, 2005.

[15] T. Brick and M. Scheutz, "Incremental natural language processing for HRI," in *Proceeding of the ACM/IEEE international conference on Human-Robot Interaction (HRI'07)*, 2007, pp. 263 – 270.

[16] K. Weilhammer, M. N. Stuttle, and S. Young, "Bootstrapping language models for dialogue systems," in *Proceedings of INTERSPEECH 2006*, Pittsburgh, PA, 2006.

[17] L. S. Zettlemoyer and M. Collins, "Online learning of relaxed CCG grammars for parsing to logical form," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 678–687.

[18] M. Collins, "Parameter estimation for statistical parsing models: theory and practice of distribution-free methods," in *New developments in parsing technology*. Kluwer Academic Publishers, 2004, pp. 19–55.

[19] S. Clark and J. R. Curran, "Log-linear models for wide-coverage ccg parsing," in *Proceedings of the 2003 conference on Empirical methods in natural language processing*. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 97–104.

[20] P. Lison and G.-J. M. Kruijff, "Salience-driven contextual priming of speech recognition for human-robot interaction," in *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI'2008)*, Patras (Greece), 2008.