An Integrated Approach to Robust Processing of Situated Spoken Dialogue

Pierre Lison
Language Technology Lab,
DFKI GmbH,
Saarbrücken, Germany
pierre.lison@dfki.de

Geert-Jan M. Kruijff
Language Technology Lab,
DFKI GmbH,
Saarbrücken, Germany
gj@dfki.de

Abstract

Spoken dialogue is notoriously hard to process with standard NLP technologies. Natural spoken dialogue is replete with disfluent, partial, elided or ungrammatical utterances, all of which are very hard to accommodate in a dialogue system. Furthermore, speech recognition is known to be a highly error-prone task, especially for complex, open-ended discourse domains. The combination of these two problems – ill-formed and/or misrecognised speech inputs – raises a major challenge to the development of robust dialogue systems.

We present an integrated approach for addressing these two issues, based on a incremental parser for Combinatory Categorial Grammar. The parser takes word lattices as input and is able to handle ill-formed and misrecognised utterances by selectively relaxing its set of grammatical rules. The choice of the most relevant interpretation is then realised via a discriminative model augmented with contextual information. The approach is fully implemented in a dialogue system for autonomous robots. Evaluation results on a Wizard of Oz test suite demonstrate very significant improvements in accuracy and robustness compared to the baseline.

1 Introduction

Spoken dialogue is often considered to be one of the most natural means of interaction between a human and a robot. It is, however, notoriously hard to process with standard language processing technologies. Dialogue utterances are often incomplete or ungrammatical, and may contain numerous disfluencies like fillers (err, uh, mm), repetitions, self-corrections, etc. Rather than getting crisp-and-clear commands such as "*Put the red ball inside the box!*", it is more likely the robot will hear such kind of utterance: "*right, now, could you, uh, put the red ball, yeah, inside the ba/ box!*". This is natural behaviour in human-human interaction (Fernández and Ginzburg, 2002) and can also be observed in several domain-specific corpora for human-robot interaction (Topp et al., 2006).

Moreover, even in the (rare) case where the utterance is perfectly well-formed and does not contain any kind of disfluencies, the dialogue system still needs to accomodate the various speech recognition errors thay may arise. This problem is particularly acute for robots operating in real-world noisy environments and deal with utterances pertaining to complex, open-ended domains.

The paper presents a new approach to address these two difficult issues. Our starting point is the work done by Zettlemoyer and Collins on parsing using relaxed CCG grammars (Zettlemoyer and Collins, 2007) (ZC07). In order to account for natural spoken language phenomena (more flexible word order, missing words, etc.), they augment their grammar framework with a small set of non-standard combinatory rules, leading to a *relaxation* of the grammatical constraints. A discriminative model over the parses is coupled with the parser, and is responsible for selecting the most likely interpretation(s) among the possible ones.

In this paper, we extend their approach in two important ways. First, ZC07 focused on the treatment of ill-formed input, and ignored the speech recognition issues. Our system, to the contrary, is able to deal with both ill-formed and misrecognized input, in an integrated fashion. This is done by augmenting the set of non-standard combinators with new rules specifically tailored to deal with speech recognition errors.

Second, the only features used by ZC07 are syntactic features (see 3.4 for details). We significantly extend the range of features included in the

Proceedings of EACL 2009 Workshop on Semantic Representation of Spoken Language - SRSL 2009, pages 58–65,
Athens, Greece, 30 March 2009. ©2009 Association for Computational Linguistics

discriminative model, by incorporating not only *syntactic*, but also *acoustic*, *semantic* and *contextual* information into the model.

An overview of the paper is as follows. We first describe in Section 2 the cognitive architecture in which our system has been integrated. We then discuss the approach in detail in Section 3. Finally, we present in Section 4 the quantitative evaluations on a WOZ test suite, and conclude.

2 Architecture

The approach we present in this paper is fully implemented and integrated into a cognitive architecture for autonomous robots. A recent version of this system is described in (Hawes et al., 2007). It is capable of building up visuo-spatial models of a dynamic local scene, 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.

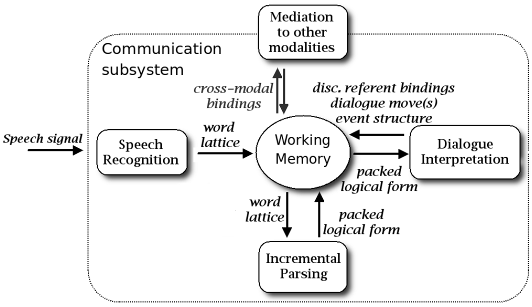

Figure 1: Architecture schema of the communication subsystem (only for comprehension).

Figure 2 illustrates the architecture schema for the communication subsystem incorporated in the cognitive architecture (only the comprehension part is shown).

Starting with 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 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 (Steedman and Baldridge, 2009). These meaning representations are ontologically richly sorted, relational structures, formulated in a (propositional) description logic, more precisely in the HLDS formalism (Baldridge and Kruijff, 2002). The parser compacts all meaning representations into a single *packed logical form* (Carroll and Oepen, 2005; Kruijff et al., 2007). A packed LF represents content similar across the different analyses as a single graph, using over- and underspecification of how different nodes can be connected to capture lexical and syntactic forms of ambiguity.

At the level of dialogue interpretation, a packed logical form is resolved against a SDRS-like dialogue model (Asher and Lascarides, 2003) to establish contextual co-reference 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* across modalities (Jacobsson et al., 2008).

2.1 Context-sensitivity

The combinatorial nature of language provides virtually unlimited ways in which we can communicate meaning. This, of course, raises the question of how precisely an utterance should then be understood as it is being heard. Empirical studies have investigated what information humans use when comprehending spoken utterances. An important observation is that interpretation *in context* plays a crucial role in the comprehension of utterance as it unfolds (Knoeferle and Crocker, 2006). During utterance comprehension, humans combine linguistic information with scene understanding and "world knowledge".

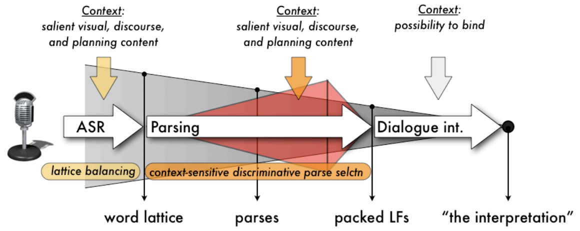
Context:
salient visual, discourse, and planning content
Context:
salient visual, discourse, and planning content
Context:
possibility to bind
ASR
Parsing
Dialogue int.
lattice balancing
context-sensitive discriminative parse selctn
word lattice
parses
packed LFs
"the interpretation"

Figure 2: Context-sensitivity in processing situated dialogue understanding

Several approaches in situated dialogue for human-robot interaction have made similar obser-

[1] Built on top of the OpenCCG NLP library: http://openccg.sf.net

vations (Roy, 2005; Roy and Mukherjee, 2005; Brick and Scheutz, 2007; Kruijff et al., 2007): A robot's understanding can be improved by relating utterances to the situated context. As we will see in the next section, by incorporating contextual information into our model, our approach to robust processing of spoken dialogue seeks to exploit this important insight.

3 Approach

3.1 Grammar relaxation

Our approach to robust processing of spoken dialogue rests on the idea of **grammar relaxation**: the grammatical constraints specified in the grammar are "relaxed" to handle slightly ill-formed or misrecognised utterances.

Practically, the grammar relaxation is done via the introduction of *non-standard CCG rules* (Zettlemoyer and Collins, 2007). In Combinatory Categorial Grammar, the rules are used to assemble categories to form larger pieces of syntactic and semantic structure. The standard rules are application ($<, >$), composition ($\mathbf{B}$), and type raising ($\mathbf{T}$) (Steedman and Baldridge, 2009).

Several types of non-standard rules have been introduced. We describe here the two most important ones: the *discourse-level composition rules*, and the *ASR correction rules*. We invite the reader to consult (Lison, 2008) for more details on the complete set of grammar relaxation rules.

3.1.1 Discourse-level composition rules

In natural spoken dialogue, we may encounter utterances containing several independent "chunks" without any explicit separation (or only a short pause or a slight change in intonation), such as

(1) "yes take the ball no the other one on your left right and now put it in the box."

Even if retrieving a fully structured parse for this utterance is difficult to achieve, it would be useful to have access to a list of smaller "discourse units". Syntactically speaking, a discourse unit can be any type of saturated atomic categories - from a simple discourse marker to a full sentence.

The type raising rule $\mathbf{T}_{du}$ allows the conversion of atomic categories into discourse units:

$$\mathsf{A} : @_i f \Rightarrow \mathsf{du} : @_i f \qquad (\mathbf{T}_{du})$$

where A represents an arbitrary saturated atomic category (s, np, pp, etc.).

The rule $>_C$ is responsible for the integration of two discourse units into a single structure:

$$\begin{aligned} &\mathsf{du} : @_i f,\ \ \mathsf{du} : @_j g \Rightarrow \\ &\qquad \mathsf{du} : @_{\{d:\text{d-units}\}}(\mathbf{list} \wedge \\ &\qquad\qquad (\langle \textsc{first} \rangle\ i \wedge f) \wedge \\ &\qquad\qquad (\langle \textsc{next} \rangle\ j \wedge g)) \quad (>_C) \end{aligned}$$

3.1.2 ASR error correction rules

Speech recognition is a highly error-prone task. It is however possible to partially alleviate this problem by inserting new error-correction rules (more precisely, new lexical entries) for the most frequently misrecognised words.

If we notice e.g. that the ASR system frequently substitutes the word "wrong" for the word "round" during the recognition (because of their phonological proximity), we can introduce a new lexical entry in the lexicon in order to correct this error:

$$round \vdash \mathsf{adj} : @_{attitude}(\mathbf{wrong}) \qquad (2)$$

A set of thirteen new lexical entries of this type have been added to our lexicon to account for the most frequent recognition errors.

3.2 Parse selection

Using more powerful grammar rules to relax the grammatical analysis tends to increase the number of parses. We hence need a a mechanism to discriminate among the possible parses. The task of selecting the most likely interpretation among a set of possible ones is called *parse selection*. Once all the possible parses for a given utterance are computed, they are subsequently filtered or selected in order to retain only the most likely interpretation(s). This is done via a (discriminative) statistical model covering a large number of features.

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

1. A function $\mathbf{GEN}(x)$ which enumerates all possible parses for an input x. In our case, this function simply represents the set of parses of x which are admissible according to 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 acoustic, syntactic, semantic or contextual features which can be relevant in discriminating the parses.

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:

$$F(x) = \operatorname*{argmax}_{y \in \mathbf{GEN}(x)} \mathbf{w}^T \cdot \mathbf{f}(x, y) \qquad (3)$$

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 *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.

3.3 Learning

3.3.1 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 (Weilhammer et al., 2006) and *generated* a corpus from a hand-written task grammar.

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

Once the grammar is specified, it is randomly traversed a large number of times, resulting in a larger set (about 25.000) of utterances along with their semantic representations. Since we are interested in handling errors arising from speech recognition, we also need to "simulate" the most frequent recognition errors. To this end, we *synthesise* each string generated by the domain-specific CFG grammar, 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[3].

3.3.2 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 and updates $\mathbf{w}$ if necessary. Albeit simple, the algorithm has proven to be very efficient and accurate for the task of parse selection (Collins and Roark, 2004; Collins, 2004; Zettlemoyer and Collins, 2005; Zettlemoyer and Collins, 2007).

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:

$$\mathbf{w} = \mathbf{w} + \mathbf{f}(x_i, y^*) - \mathbf{f}(x_i, y') \qquad (4)$$

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

The most expensive step in this algorithm is the calculation of $y' = \operatorname{argmax}_{y \in \mathbf{GEN}(x_i)} \mathbf{w}^T \cdot \mathbf{f}(x_i, y)$ - this is the *decoding* problem.

It is possible to prove that, provided the training set (x_i, z_i) is separable with margin $\delta > 0$, the algorithm is assured to converge after a finite number of iterations to a model with zero training errors (Collins and Roark, 2004). See also (Collins, 2004) for convergence theorems and proofs.

3.4 Features

As we have seen, the parse selection operates by enumerating the possible parses and selecting the

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

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

Algorithm 1 Online perceptron learning

```
Require: - set of $n$ training examples $\{(x_i, z_i) : i = 1...n\}$
         - $T$: 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
$\mathbf{w} \leftarrow \mathbf{w_0}$

% Loop $T$ times on the training examples
for $t = 1...T$ do
    for $i = 1...n$ do
        % Compute best parse according to current model
        Let $y' = \text{argmax}_{y \in \mathbf{GEN}(x_i)} \mathbf{w}^T \cdot \mathbf{f}(x_i, y)$
        % If the decoded parse $\neq$ expected parse, update the parameters
        if $L(y') \neq z_i$ then
            % Search the best parse for utterance $x_i$ with semantics $z_i$
            Let $y^* = \text{argmax}_{y \in \mathbf{GEN}(x_i, z_i)} \mathbf{w}^T \cdot \mathbf{f}(x_i, y)$
            % Update parameter vector $\mathbf{w}$
            Set $\mathbf{w} = \mathbf{w} + \mathbf{f}(x_i, y^*) - \mathbf{f}(x_i, y')$
        end if
    end for
end for
return parameter vector $\mathbf{w}$
```

one with the highest score according to the linear model parametrised by $\mathbf{w}$.

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. They must also be relatively cheap to compute. In our model, the features are of four types: semantic features, syntactic features, contextual features, and speech recognition features.

3.4.1 Semantic features

What are the substructures of a logical form which may be relevant to discriminate the parses? We define features on the following information sources:

1. *Nominals*: for each possible pair $\langle prop, sort\rangle$, we include a feature f_i in $\mathbf{f}(x, y)$ counting the number of nominals with ontological sort *sort* and proposition *prop* in the logical form.

2. *Ontological sorts*: occurrences of specific ontological sorts in the logical form.

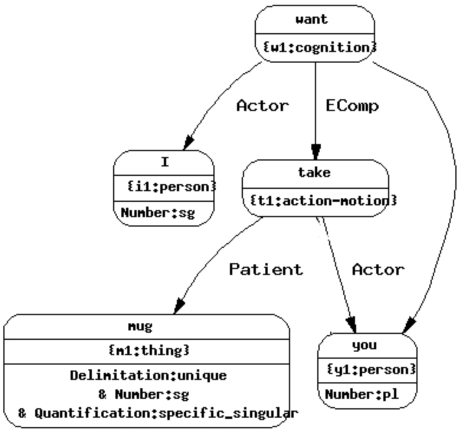

Figure 3: graphical representation of the HLDS logical form for "*I want you to take the mug*".

3. *Dependency relations*: following (Clark and Curran, 2003), we also model the *dependency structure* of the logical form. Each dependency relation is defined as a triple $\langle sort_a,\ sort_b,\ label\rangle$, where $sort_a$ denotes the sort of the incoming nominal, $sort_b$ the sort of the outgoing nominal, and $label$ is the relation label.

4. *Sequences of dependency relations*: number of occurrences of particular sequences (ie. bigram counts) 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 lexical and syntactic ambiguities.

3.4.2 Syntactic features

By "syntactic features", we mean features associated to the *derivational history* of a specific parse. The main use of these features is to *penalise* to a correct extent the application of the non-standard rules introduced into the grammar.

To this end, we include in the feature vector $\mathbf{f}(x, y)$ a new feature for each non-standard rule, which counts the number of times the rule was applied in the parse.

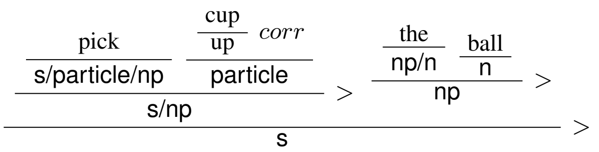

Figure 4: CCG derivation of *"pick cup the ball"*.

In the derivation shown in the figure 4, the rule $corr$ (correction of a speech recognition error) is applied once, so the corresponding feature value is set to 1. The feature values for the remaining rules are set to 0, since they are absent from the parse.

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 mechanism ensures that the grammar relaxation is only applied "as a last resort" when the usual grammatical analysis fails to provide a full parse. Of course, depending on the relative frequency of occurrence of these rules in the training corpus, some of them will be more strongly penalised than others.

3.4.3 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:

1. *Activated words*: our dialogue system maintains in its working memory a list of contextually activated words (cfr. (Lison and Kruijff, 2008)). 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.

2. *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.

3. *Expected syntactic categories*: for each atomic syntactic category in the CCG grammar, we include one feature indicating if the category is consistent with the current discourse model. These features can be used to handle *sentence fragments*.

3.4.4 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 of such a structure is given in Figure 5. Each recognition hypothesis is provided with an associated confidence score, and 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 three features: the *acoustic confidence score* (confidence score provided by the statistical models included in the ASR), the *semantic confidence score* (based on a "concept model" also provided by the ASR), and the *ASR ranking* (hypothesis rank in the word lattice, from best to worst).

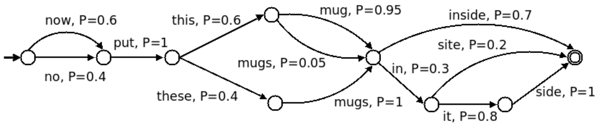

Figure 5: Example of word lattice

4 Experimental evaluation

We performed a quantitative evaluation of our approach, using its implementation in a fully integrated system (cf. Section 2). To set up the experiments for the evaluation, we have gathered a corpus of human-robot spoken dialogue for our task-domain, which we segmented and annotated manually with their expected semantic interpretation. The data set contains 195 individual utterances along with their complete logical forms.

4.1 Results

Three types of quantitative results are extracted from the evaluation results: *exact-match*, *partial-match*, and *word error rate*. Tables 1, 2 and 3 illustrate the results, broken down by use of grammar relaxation, use of parse selection, and number of recognition hypotheses considered.

	Size of word lattice (number of NBests)	Grammar relaxation	Parse selection	Precision	Recall	F_1-value
(Baseline)	1	No	No	40.9	45.2	**43.0**
.	1	No	Yes	59.0	54.3	56.6
.	1	Yes	Yes	52.7	70.8	60.4
.	3	Yes	Yes	55.3	82.9	66.3
.	5	Yes	Yes	55.6	84.0	66.9
(Full approach)	10	Yes	Yes	55.6	84.9	**67.2**

Table 1: Exact-match accuracy results (in percents).

	Size of word lattice (number of NBests)	Grammar relaxation	Parse selection	Precision	Recall	F_1-value
(Baseline)	1	No	No	86.2	56.2	**68.0**
.	1	No	Yes	87.4	56.6	68.7
.	1	Yes	Yes	88.1	76.2	81.7
.	3	Yes	Yes	87.6	85.2	86.4
.	5	Yes	Yes	87.6	86.0	86.8
(Full approach)	10	Yes	Yes	87.7	87.0	**87.3**

Table 2: Partial-match accuracy results (in percents).

Each line in the tables corresponds to a possible configuration. Tables 1 and 2 give the precision, recall and F_1 value for each configuration (respectively for the exact- and partial-match), and Table 3 gives the Word Error Rate [WER].

The first line corresponds to the baseline: no grammar relaxation, no parse selection, and use of the first NBest recognition hypothesis. The last line corresponds to the results with the full approach: grammar relaxation, parse selection, and use of 10 recognition hypotheses.

Size of word lattice (NBests)	Grammar relaxation	Parse selection	WER
1	No	No	**20.5**
1	Yes	Yes	19.4
3	Yes	Yes	16.5
5	Yes	Yes	15.7
10	Yes	Yes	**15.7**

Table 3: Word error rate (in percents).

4.2 Comparison with baseline

Here are the comparative results we obtained:

- Regarding the exact-match results between the baseline and our approach (grammar relaxation and parse selection with all features activated for NBest 10), the F_1-measure climbs from 43.0 % to 67.2 %, which means a relative difference of $\mathbf{56.3}$ %.

- For the partial-match, the F_1-measure goes from 68.0 % for the baseline to 87.3 % for our approach – a relative increase of $\mathbf{28.4}$ %.

- We obverse a significant decrease in WER: we go from 20.5 % for the baseline to 15.7 % with our approach. The difference is statistically significant (p-value for t-tests is 0.036), and the relative decrease of $\mathbf{23.4}$ %.

5 Conclusions

We presented an *integrated* approach to the processing of (situated) spoken dialogue, suited to the specific needs and challenges encountered in human-robot interaction.

In order to handle disfluent, partial, ill-formed or misrecognized utterances, the grammar used by the parser is "relaxed" via the introduction of a set of *non-standard combinators* which allow for the insertion/deletion of specific words, the combination of discourse fragments or the correction of speech recognition errors.

The relaxed parser yields a (potentially large) set of parses, which are then packed and retrieved by the parse selection module. The parse selection is based on a discriminative model exploring a set of relevant semantic, syntactic, contextual and acoustic features extracted for each parse. The parameters of this model are estimated against an automatically generated corpus of $\langle$utterance, logical form$\rangle$ pairs. The learning algorithm is an perceptron, a simple albeit efficient technique for parameter estimation.

As forthcoming work, we shall examine the potential extension of our approach in new directions, such as the exploitation of parse selection for *incremental* scoring/pruning of the parse chart, the introduction of more refined contextual features, or the use of more sophisticated learning algorithms, such as Support Vector Machines.

References

Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Cambridge University Press.

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

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

J. Carroll and S. Oepen. 2005. High efficiency realization for a wide-coverage unification grammar. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP'05)*, pages 165–176.

Stephen Clark and James R. Curran. 2003. Log-linear models for wide-coverage ccg parsing. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 97–104, Morristown, NJ, USA. Association for Computational Linguistics.

Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *ACL '04: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, page 111, Morristown, NJ, USA. Association for Computational Linguistics.

Michael Collins. 2004. Parameter estimation for statistical parsing models: theory and practice of distribution-free methods. In *New developments in parsing technology*, pages 19–55. Kluwer Academic Publishers.

R. Fernández and J. Ginzburg. 2002. A corpus study of non-sentential utterances in dialogue. *Traitement Automatique des Langues*, 43(2):12–43.

N. Hawes, A. Sloman, J. Wyatt, M. Zillich, H. Jacobsson, G.J. M. Kruijff, M. Brenner, G. Berginc, and D. Skocaj. 2007. Towards an integrated robot with multiple cognitive functions. In *AAAI*, pages 1548–1553. AAAI Press.

Henrik Jacobsson, Nick Hawes, Geert-Jan Kruijff, and Jeremy Wyatt. 2008. Crossmodal content binding in information-processing architectures. In *Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Amsterdam, The Netherlands, March 12–15.

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

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

Pierre Lison and Geert-Jan M. Kruijff. 2008. Salience-driven contextual priming of speech recognition for human-robot interaction. In *Proceedings of the 18th European Conference on Artificial Intelligence*, Patras (Greece).

Pierre Lison. 2008. Robust processing of situated spoken dialogue. Master's thesis, Universität des Saarlandes, Saarbrücken.

D. Roy and N. Mukherjee. 2005. Towards situated speech understanding: visual context priming of language models. *Computer Speech & Language*, 19(2):227–248, April.

D.K. Roy. 2005. Semiotic schemas: A framework for grounding language in action and perception. *Artificial Intelligence*, 167(1-2):170–205.

Mark Steedman and Jason Baldridge. 2009. Combinatory categorial grammar. In Robert Borsley and Kersti Börjars, editors, *Nontransformational Syntax: A Guide to Current Models*. Blackwell, Oxford.

E. A. Topp, H. Hüttenrauch, H.I. Christensen, and K. Severinson Eklundh. 2006. 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.

Karl Weilhammer, Matthew N. Stuttle, and Steve Young. 2006. Bootstrapping language models for dialogue systems. In *Proceedings of INTERSPEECH 2006*, Pittsburgh, PA.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI '05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, July 2005*, pages 658–666.

Luke Zettlemoyer and Michael Collins. 2007. 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)*, pages 678–687.