Semantic parsing of speech using grammars learned with weak supervision

Judith Gaspers
Semantic Computing Group
CITEC
Bielefeld University

Philipp Cimiano
Semantic Computing Group
CITEC
Bielefeld University

Britta Wrede
Applied Informatics
Faculty of Technology
Bielefeld University

{jgaspers|cimiano|bwrede}@cit-ec.uni-bielefeld.de

Abstract

Semantic grammars can be applied both as a language model for a speech recognizer and for semantic parsing, e.g. in order to map the output of a speech recognizer into formal meaning representations. Semantic speech recognition grammars are, however, typically created manually or learned in a supervised fashion, requiring extensive manual effort in both cases. Aiming to reduce this effort, in this paper we investigate the induction of semantic speech recognition grammars under weak supervision. We present empirical results, indicating that the induced grammars support semantic parsing of speech with a rather low loss in performance when compared to parsing of input without recognition errors. Further, we show improved parsing performance compared to applying n-gram models as language models and demonstrate how our semantic speech recognition grammars can be enhanced by weights based on occurrence frequencies, yielding an improvement in parsing performance over applying unweighted grammars.

1 Motivation

Semantic parsers map natural language utterances (*NL*) into formal meaning representations (*MR*), and are applied for both parsing of textual input and in Spoken Language Understanding (SLU). In data-driven SLU research, typically pipeline-based systems are applied in which first an automatic speech recognizer (ASR) is applied to transcribe speech input, and subsequently a semantic parser is applied to map the transcriptions into some semantic form (Deoras et al., 2013). Such systems typically use different models for recognition and understanding. Since ASR yields recognition errors, parsing performance can degrade rapidly compared to parsing performance on written text. While the performance of ASR and parsing components are often optimized independently of each other, in particular in case of the ASR to minimize recognitions errors, research has shown that ASR transcriptions with a lower error rate can in fact yield worse understanding performance (Wang et al., 2003; Bayer and Riccardi, 2012) and that joint approaches to recognition and understanding can yield improved performance (Wang and Acero, 2006b; Deoras et al., 2013). In particular, Wang and Acero (2006b) have shown that applying the same grammar for speech recognition and understanding can yield improved understanding performance compared to applying a standard n-gram model with the ASR, since dependencies between acoustics and semantics can be captured. Their grammars are, however, learned in a supervised setting. In fact, while semantic grammars are often applied for speech recognition and/or understanding, they are often created manually or – as mentioned previously – learned from data containing semantic annotations, which are time-consuming to produce.

In the field of Natural Language Processing (NLP), the development of semantic parsers has received considerable attention. While some researchers have considered fully supervised settings (Wong and Mooney, 2006; Zettlemoyer and Collins, 2007), requiring accurate and complete semantic annota-

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 872–881,
Denver, Colorado, May 31 – June 5, 2015.

tions, others have developed weakly supervised approaches exploiting ambiguous representations of the context in which an utterance is produced instead of accurate and complete annotations (Chen et al., 2010; Börschinger et al., 2011; Chen and Mooney, 2008). In this line, in this paper we explore how an approach that induces semantic parsers in the form of a (semantic) grammar from ambiguous training data can be applied to acquire a language model (LM) for speech recognition as well as a semantic parser for the understanding task at the same time. Making use of a semantic parser as a language model for speech recognition also comes with the advantage that no separate language model must be trained. In our experiments, we compare performance of the induced grammars to the performance of different language models, in particular n-gram models, and we investigate the impact of enhancing induced semantic grammars with weights based on the training data. We present empirical results showing that it is possible to induce semantic grammars with weak supervision that can be applied successfully both as an LM for a speech recognizer and for semantic parsing. We show that with respect to parsing performance, our joint approach in which the same grammar is used for parsing and as an LM yields a higher F_1 (84.46%) compared to an approach in which a standard n-gram based model is used as an LM (78.36%). In addition, our results indicate that enhancing speech recognition grammar rules with weights based on occurrence frequencies can yield improved performance over unweighted grammars (84.46% vs 82.37% for weighted vs unweighted grammars, respectively).

2 Background & related work

In principle, two different types of language models can be applied with an ASR: stochastic LMs – typically n-gram models – and speech recognition grammars. While n-gram models estimate probabilities of word sequences, speech recognition grammars explicitly specify rules defining which words and patterns a user may utter. Further, semantic information can be directly included within the rules. Thus, when applied with an ASR, spoken utterances can be directly transformed into a corresponding semantic representation without producing a sequence of words as intermediate step. This approach is typically taken when building commercial systems (Wang et al., 2011). Such grammars are, however, typically created manually, which is time-consuming and error-prone. Hence, data-driven approaches to automatic grammar induction have been explored (Wang and Acero, 2006b; Wang and Acero, 2005; Wang and Acero, 2003). However, they often rely on fully supervised settings, requiring training data which is annotated at the utterance- or word level, which is costly and time-consuming to produce. In contrast, aiming to reduce the required manual effort, in this paper we explore the utility of weak supervision in the form of ambiguous context information for the induction of grammars applicable for both speech recognition and understanding.

The utility of this kind of weak supervision has been explored previously in the field of semantic parsing (Chen et al., 2010; Börschinger et al., 2011; Chen and Mooney, 2008), and unsupervised approaches to semantic parsing have been proposed as well (Poon and Domingos, 2009; Goldwasser et al., 2011). While such approaches may be applied as parsing components for SLU systems – notice though that the SLU task differs from parsing of written text in that recognition errors and phenomena of spoken language must be handled, and that not all SLU models can be applied as an LM (Wang et al., 2011) – we are not aware of work aiming to transform these parsers into speech recognition grammars or investigating their performance with respect to different LMs applied with an ASR.

Semantic parsers applied in pipeline-based SLU systems are in general usually learned in a supervised fashion. Other than semantic grammar-based approaches, probabilistic models and machine learning techniques have been applied in SLU for conceptual tagging due to their robustness to noise, e.g. Conditional Random Fields (Lafferty et al., 2001) have been applied (e.g. Wang and Acero (2006a; Dinarelli et al. (2012)); He and Young (2005) present an approach based on Hidden Markos Models. However, evaluations have shown that even in case of applying machine learning techniques or probabilistic models, semantic parsing of ASR transcriptions is affected by much more errors compared to parsing of correct transcriptions (De Mori, 2011). In order to reduce annotation costs, work has,

for instance, focused on providing annotation tools (Wang and Acero, 2006b; Wang and Acero, 2005), exploring supervised learning in combination with active learning (Wu et al., 2010) and gaining additional training data, for instance, from the Web using queries generated from a (small) existing grammar (Klasinas et al., 2013). These approaches, however, still assume manual effort and may be somewhat complementary to the one investigated here. Further, data-driven SLU parsers are often based on rather local features, e.g. n-grams, while we explore template-based grammars which can capture long-distance linguistic dependencies.

Several approaches have addressed unsupervised (Solan et al., 2005; van Zaanen and Adriaans, 2001) and semi-supervised (Wong and Meng, 2001; Siu and Meng, 1999; Meng and Siu, 2002) induction of grammars, where the latter may comprise manual post-processing of automatically induced rules. In particular, in order to be applicable as an SLU model, semantic information must be added manually, since only syntactic structures can be induced automatically in this case.

While in data-driven SLU research typically pipeline-based systems are applied, a few joint approaches have been proposed (Deoras et al., 2013; Wang and Acero, 2006b; Bayer and Riccardi, 2012). Specifically, the work presented here is most similar to the approach presented by Wang and Acero (2006b). In particular, we also attempt to learn grammars applicable for both speech recognition and understanding. However, Wang and Acero (2006b) explore a supervised setting based on word-level annotations for slots and induce rather local rules, i.e. based on preambles and postambles for slots, while we explore a template-based approach, capturing long-distance linguistic dependencies.

3 Methodology

In this paper, we explore the induction of semantic grammars under weak supervision provided in the form of ambiguous representations of the semantic context as explored in the NLP field of Semantic Parsing (Chen et al., 2010). In particular, the training data comprises of a set of textual utterances coupled with symbolic context information from which we induce semantic parsers and derive different LMs for application with an ASR. LMs are then applied to transcribe speech data, and the resulting transcriptions are in turn mapped into meaning representations by the learned semantic parsers. In the following, we will first describe the input data and learning scenario and subsequently the semantic parsing approach as well as the creation of language models.

3.1 Learning scenario and input data

Our experiments were performed on the RoboCup soccer corpus (Chen and Mooney, 2008), which is a standard dataset used for the evaluation of semantic parsing algorithms taking written natural language utterances as input. The corpus comprises four RoboCup games. Game events are represented by predicate logic formulas, which represent the ambiguous contextual representations from which semantic parsers are trained in a weakly supervised fashion. The games were commented by humans, yielding examples for *written* natural language utterances (*NL*). In the corpus, each *NL* is paired with a set of possible meaning representations $mr_i \in MR$, each expressing a game action, and *NL* corresponds to at most one them. For example, *pass(purple10,purple7)* represents an *mr* for a passing event which might be commented as "purple10 kicks to purple7". However, there is no direct correspondence between the *NL* comments and their corresponding *mr*s; thus, these correspondences have to be learned.

The corpus also contains a gold standard comprising *NL*s annotated with their correct *mr*s. Several semantic parsers have been evaluated using this dataset by applying the evaluation schema introduced by Chen et al. (2010). The authors performed 4-fold cross-validation on the four games. Training was done on the ambiguous training data, while the gold standard for a fourth game was used for testing. Results were presented by means of the F_1 score. Precision and recall were computed as the percentage of *mr*s produced by the system that were correct and the percentage of *mr*s that the system produced correctly, respectively. A parse was considered as correct if it matched the gold standard exactly (Chen et al., 2010). Recently, this task has been extended to consider speech data, both in learning and applying a parser. In particular, the approach of Gaspers and Cimiano (2014) relied on transcriptions made by a

task-independent phoneme recognizer as input. For this purpose, *NL* comments contained in the dataset were read by a speaker. By contrast, in this paper we explore how grammars for speech recognition and understanding can be built from textual input in a weakly supervised setting, and subsequently be applied for recognition and parsing of speech input. Hence, we explore a 4-fold cross-validation scenario in which for each fold learning is performed using the *written* ambiguous training data for three games, while the *spoken* gold standard of the forth game is used for testing, i.e. for performing both speech recognition and subsequent parsing of the resulting ASR transcriptions; spoken data are the same as in Gaspers and Cimiano (2014). For application with the ASR we normalized training data which mainly comprised lowercasing and replacement of numbers in player names, e.g. "pink4" → "pink four". Some statistics for the normalized dataset are presented in Table 1.[1]

Table 1: Dataset statistics.

Total number of comments	1,872
Comments having correct mr	1,539
Average number of events per comment	2.5
Maximum number of events per comment	12
SD in number of events per comment	1.8
Mean utterance length	7.39
# Types	355
# Tokens	13,838

3.2 The applied semantic parsing algorithm

For semantic parser induction we applied the algorithm presented in Gaspers and Cimiano (2014), which is mainly designed to work with the output of a phoneme recognizer. The algorithm is also applicable to textual input and has been shown to achieve state-of-the-art performance on written input (cf. Gaspers and Cimiano (2014)). The induced parser is represented in the form of a lexicon and an inventory containing syntactic constructions and thus well-suited to be transformed into a rule-based speech recognition grammar. The learned lexicon comprises lexical units, i.e. words or short sequences of words, along with their mapping to semantic referents, e.g. "pink goalie" → $pink1$. Each syntactic construction consists of a syntactic pattern, e.g. "X_1 kicks to X_2", along with an associated semantic frame, e.g. $pass(ARG_1, ARG_2)$, and a mapping which maps slots in the syntactic pattern to argument slots in the semantic frame, e.g. $X_1 \rightarrow ARG_1, X_2 \rightarrow ARG_2$. Slots in syntactic patterns represent positions in which a lexical unit from the parser's lexicon can be inserted. For instance, in the previous pattern "pink goalie" can be inserted at position X_1 or X_2. When applied to written text, parser induction is performed by applying the following learning steps:

1. acquisition of an initial lexicon
2. computation of alignments between *NL*s and ambiguous context representations, and
3. estimation of co-occurrence frequencies at different levels.

This work flow is illustrated in Fig 1.

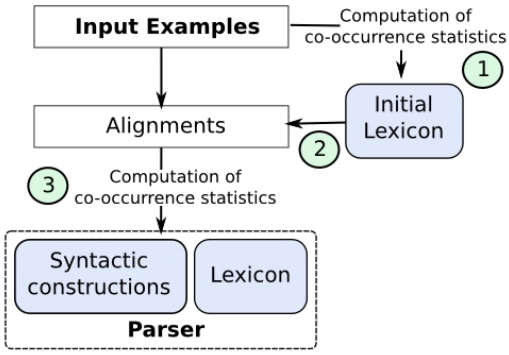

Figure 1: The algorithm's work flow.

In step 1, initial lexical knowledge is learned by computing co-occurrence frequencies between all bi- and unigrams appearing in the *NL* data and all semantic referents appearing in the *MR* data. In step 2, this knowledge is used to compute alignments for each example between its *NL* and all $mr_i \in MR$ observed with it, i.e. lexical knowledge is used to segment *NL* such that all semantic referents observed in an *mr* are expressed by individual sequences, and hypotheses concerning a mapping between *NL* and

[1] Numbers for mean utterance length and number of tokens and types are computed only for comments included in the training dataset. Regarding the total number of comments we use one more per game than Chen et al. (2010) in line with Börschinger et al. (2011).

semantics are created. For instance, given an input example

(1)

NL:	purple eight kicks to purple seven
mr_1:	$playmode(play\_on)$
mr_2:	$pass(purple8, purple7)$
mr_3:	$pass(purple2, purple5)$

the following alignment might be created:

(2)

NL	X_1 kicks to X_2
mr	$pass(ARG_1, ARG_2)$ Mapping: $X_1 \rightarrow ARG_1$, $X_2 \rightarrow ARG_2$
$nl \rightarrow ref$	purple eight $\rightarrow purple8$ purple seven $\rightarrow purple7$

That is, assuming that the algorithm has learned in step 1 that "purple eight" and "purple seven" refer to $purpl8$ and $purple7$, respectively, it may use this knowledge to hypothesize that *NL* is an instantiation of a pattern "X_1 kicks to X_2", which in turn refers to the predicate $pass(ARG_1, ARG_2)$ with a mapping $X_1 \rightarrow ARG_1$, $X_2 \rightarrow ARG_2$.
Alignments are rated, and for each example only those having maximal scores are used for parser induction. By this, lexical knowledge is used to predisambiguate the training data. Given the alignments induced in step 3, a parser is estimated by computing co-occurrence frequencies at different levels. In particular, association scores are computed at three different levels, i.e.

1. $nl \rightarrow ref$: between all lexical units, e.g. "purple eight", and semantic referents, e.g. *purple8*, appearing in alignments,

2. *NL* $\rightarrow mr$: between all syntactic patterns, e.g. "X_1 kicks to X_2", and semantic frames, e.g. $pass(ARG_1, ARG_2)$, appearing in alignments, and

3. mapping: between all slots in a syntactic pattern, e.g. X_1, and argument slots, e.g. ARG_1, specific for each pattern and semantic frame.

Then, the parser's lexicon consists of rules of the form $nl \rightarrow ref$, while the syntactic constructions have the form *NL* $\rightarrow mr$, each coupled with its individual mapping.
Parsing is performed by searching for an appropriate syntactic construction given an input *NL*, and the arguments matching the elements at the slots in the syntactic pattern are inserted into the appropriate argument slots in the associated semantic frame. Approximate matching can be applied during parsing of *NL*s for which no pattern can be found otherwise. In this paper, we always perform approximate matching by searching for a matching syntactic pattern with a Levenshtein distance of 1 if no matching pattern can be found directly, which allows us to parse utterances containing a recognition error. Even though – of course – more than one recognition error might be contained in a given utterance, we do not use greater distance values because this would likely yield parsing errors, as utterances are rather short and most of the words are important for detecting the meaning; leaving out too many words will in general increase the likelihood of matching wrong patterns, thus yielding spurious interpretations. Using the algorithm, for each fold of the RoboCup data set we created a semantic parser using the written training data of three games.

3.3 Creation of language models

Based on the written training data we created different LMs. In particular, we created rule-based recognition grammars using the algorithm and further LMs, such as trigram models, for comparison.

3.3.1 Recognition grammars

We built semantic speech recognition grammars given a semantic parser by transforming all rules with an occurrence greater than one into JSpeech Grammar Format (JSGF)[2]. The resulting grammars consisted of rules representing the parser's inventory of syntactic constructions as well as its lexicon. In case of the inventory of syntactic constructions, alternative expansions of learned syntactic patterns were defined, and in case of the lexicon, alternative expansions of learned lexical units were defined. In particular, with respect to the lexicon we defined a rule <ref> which comprises the learned lexical units. With respect to syntactic constructions we defined a rule <utterance> which comprises the patterns. Further, syntactic slots in patterns were replaced by <ref>, allowing lexical units to appear at those positions. In grammar creation, we also investigated the influence of occurrence frequencies of

[2]http://www.w3.org/TR/jsgf/

syntactic patterns and lexical units to enhance grammatical rules with weights. In particular, we created both weighted and unweighted grammars. When using weights, rules were weighted by using occurrence frequencies, i.e. the frequency which was observed for pattern or lexical unit as aligned by the algorithm during training. Hence, weights for patterns and lexical units aligned less frequently in the training data were smaller, indicating that they were less likely to be spoken. An example illustrating a subset of two (weighted) rules is illustrated in Fig. 2.

Notice that resulting JSGF grammars do not explicitly contain semantic information, but their induction was driven by semantic information. This is the case because a mapping to semantics was not needed during recognition as we explore a two-stage approach where parsing is performed after recognition, allowing the inclusion of further LMs during recognition. However, because both parsing and understanding are performed using the same grammar – where semantic information is ignored by the LM – it would also be possible to induce a semantic grammar that directly maps ASR output into semantic representations.

3.3.2 Baseline language models

We computed different language models for comparison; these were mainly stochastic LMs. In particular, we created standard trigram language models from the written training data without making use of concurrent perceptual context information using SRILM (Stolcke, 2002). Since the RoboCup corpus is rather small and n-gram models are typically learned from large amounts of data, in addition we interpolated the trigram models trained solely on the in-domain RoboCup corpus each with a large background language model trained on a broadcast news corpus, i.e. the HUB4 dataset (Fiscus et al., 1998). We also experimented with class-based models, but automatic induction of classes in an unsupervised fashion did not appear promising and we refrained from manually creating classes since the focus of this paper is on the automatic creation of ASR resources without requiring extensive manual effort. However, an interesting experiment would be to utilize the semantic classes induced by our algorithm in order to create class-based language models.

Moreover, in order to evaluate the utility of ambiguous perceptual context for speech recognition grammar induction, as a fully unsupervised grammar-based baseline, we induced syntactic grammars relying on the ADIOS algorithm (Solan et al., 2005). Notice, however, that it is not common to apply grammars learned in an unsupervised fashion directly for SLU. In particular, with respect to semantic parsing, automatically induced grammars are typically post-processed manually, which we refrained from doing, since the focus of this paper is on the automatic creation of speech recognition and understanding components.

4 Experiments & Results

We evaluated the word error rate (WER) as well as parsing accuracy for different language models and combinations thereof. In particular, in case of applying recognition grammars we applied these also in combination with an n-gram back off LM. In particular, the n-gram model was applied in case of utterances which were rejected by the recognizer as out of grammar (OOG), as these might still be parsed subsequently by applying approximate matching. Notice, however, that for our experiments we did not apply both LMs at a time but combined the output of two recognizers for further processing. Notice further that most speech recognizers can only be applied using either a recognition grammar or an n-gram model at a time, but one can assume that two recognizers might be configured to run in parallel. As mentioned previously, we performed 4-fold cross-validation on the four RoboCup games. For each fold, learning semantic parsers and creation of language models was performed using the ambiguous *written* training data for three games and the *spoken* gold standard for the forth game for testing. In the following, we will discuss results for applying our induced grammars as an LM compared to using standard trigrams models (solely trained on the in-domain data) as a baseline, since these yielded the best results. In particular, we do not discuss the results achieved by the grammars induced in a purely syntactic manner as they performed worse than semantic grammars in all experiments, and we do not discuss the experiments for the interpolated/adapted

Figure 2: A subset of weighted speech recognition grammar rules

```
public <utterance> = /6/ <ref> again passes to <ref> | /199/ <ref> kicks to <ref> | ...
<ref> =  /15/ pink goalie | /132/ pink nine | /10/ pink one | ...
```

trigram models as they performed worse than the in-domain trigram models with the exception of a very slight improvement when applied as a back off model for SLU.[3]

4.1 Speech recognition

Speech recognition was performed using different (combinations of) LMs individually; lexicon and acoustic models were the same in all cases.[4] Speech recognition results with respect to the word error rate averaged over all folds are presented in Table 2.

Table 2: speech recognition results

Applied language model(s)	**WER (%)**
Semantic grammar w/o weights	15.55
Semantic grammar w/o weights + trigram back off	12.63
Semantic grammar inc. weights	17.15
Semantic grammar inc. weights + trigram back off	10.88
Trigram (baseline)	**7.1**

As can be seen, with a rather low error rate of 7.1%, applying trigram language models yields the best results. While in case of applying semantically motivated recognition grammars the WER increases, it must be noted that in cases in which no back off models were applied this is to some extent due to OOG utterances (as these yield several deletions compared to the reference data). Yet, the OOG-rate is rather low, i.e. averaged over all folds 8.6% and 4.1% when using grammars with and without weights, respectively. However, even in cases where OOG utterances are recognized by applying trigram language models, the WER is higher compared to applying trigram language models only. Notably, these results were not consistent across folds. For two folds, the WER actually decreased when combining a semantically motivated grammar including weights with a trigram language model compared to applying the trigram language model only, thus indicating that combining semantically motivated grammars learned with weak supervision with trigram models can also yield improved recognition performance over applying trigram models only in some cases.

4.2 Semantic parsing

For each fold, ASR transcriptions were parsed using the semantic parser learned on the training data for that fold. For comparison, as an upper baseline we computed parsing performance on normalized gold standard data, since typically performance degrades – and often to a large extent – when a semantic parser is applied to ASR transcriptions of speech; recall that the applied algorithm achieves state-of-the-art performance on the dataset.[5] Results are presented in Table 3.

Table 3: Semantic parsing results on written text and on speech transcribed using different language models

Written text (reference)			
	F_1	**Prec.**	**Recall**
Normalized text	87.26	94.28	81.42
Speech			
Applied language model(s)	**F_1**	**Prec.**	**Recall**
Semantic grammar inc. weights	84.18	88.7	80.18
Semantic grammar inc. weights + trigram back off	**84.46**	87.53	81.64
Semantic grammar w/o weights	82.24	84.83	79.84
Semantic grammar w/o weights + trigram back off	82.37	84.67	80.21
Trigram (baseline)	78.36	90.34	69.4

[3]The results were: interpolated/adapted LM: WER: 13.43%, F_1: 71.22%, semantic grammar + interpolated/adapted LM backoff: WER: 13.85, F_1: 84.6%, syntactic recognition grammar: WER: 18.98%, F_1: 70.86%, syntactic recognition grammar + trigram back off: WER: 13.98%, F_1: 71.27%.

[4]We applied Sphinx4 (Walker et al., 2004) using lexicon and acoustic models trained on the HUB4 dataset (Fiscus et al., 1998), which contains broadcast news speech matching our RoboCup data with respect to acoustics in that in both cases read speech is addressed; these resources are available online. We added phonetic transcriptions for out of vocabulary (OOV) to the vocabulary; only two were OOV along with some typos.

[5]While comparison with a manually created gold standard grammar would be interesting as well, a manually created grammar for the utilized dataset is unfortunately not available.

The results reveal that in case of applying trigram LMs F_1 degrades about 9% absolute compared to parsing written text (reference), yielding 78.36% in F_1, even though the WER is rather low with a value of 7.1%. Thus, the trigram seems to "destroy" semantically meaningful sequences while restoring sequences that contain no meaning. By contrast, in case of applying a semantically motivated recognition grammar including weights, performance improves by 6% absolute over the trigram model, even though the WER is higher in this case. Moreover, including weights in the recognition grammars yields improved performance compared to using unweighted grammars.

Notably, the decrease in performance in case of applying a weighted semantically motivated recognition grammar (+ trigram back off) compared to performance on the reference data is mainly due to a decrease in precision. Here it must be noted that the high values in F_1 are achieved without performing any optimization of (recognition) parameters. In the performed experiments, the probability for OOG utterances was rather low, and thus utterances were matched incorrectly by the ASR which were actually not covered by the grammar, yielding both recognition and subsequent parsing errors. However, these parameters can be tuned, likely increasing precision and F_1 even further (and probably also the WER).

Applying a back off trigram model yields only little improvement in parsing performance, although this may to some extent be due to not tuning recognition parameters. That is, if the ASR would be tuned to reject more OOG utterances correctly, these utterances might instead be recognized by a trigram model and probably parsed correctly by applying approximate matching.

5 Discussion

When applying trigram models, even with a rather low error rate of 7.1%, semantic parsing performance degraded about 9% absolute in F_1. Here it must be noted that due to the evaluation schema a single recognition error can yield a completely incorrect parse. Recall that evaluation is performed on the basis of fully correct *mr*s, i.e. all referents and the predicate must be determined correctly in order to yield a correct parse. For instance, if any of the words "purple", "pink", "two" or "five" is deleted or substituted in an utterance "purple two passes to pink five", one of the referents may not be identified (correctly). Similarly, deleting or substituting "passes" may yield an incorrect predicate or no parse at all. Hence, parsing performance can degrade rapidly even on ASR transcriptions containing only few recognition errors.

The results show that, in line with previous research (Wang et al., 2003; Bayer and Riccardi, 2012), a lower WER may not yield better understanding results, i.e. in our case parsing performance is not directly dependent on the WER but rather on the type of errors made. In particular, with respect to semantic parsing it is important that words carrying important meaning are recognized correctly. For instance, a spoken utterance "pink nine passes the ball to pink seven" in which "seven" is incorrectly recognized as "eleven" likely yields a parsing error while a recognition error which substitutes "backward" by "forward" may not prevent correct parsing. This is the case because "forward" and "backward" do not carry any semantics in the data set at hand, while correct identification of numbers is in most cases essential for detecting the correct semantic referents. Applying the semantically motivated grammars may have been beneficial in recognizing the semantic referents correctly because the system can explicitly learn them and their appearances in certain patterns in contrast to the trigram model. In particular, if an utterance "pink nine passes the ball to pink seven" appears during recognition and "pink seven" has not been observed in the context of the preceding words during training, then the n-gram model would assign a low probability, likely leading to a recognition error such as "pink eleven". By contrast, in case of semantic grammars the system can learn that the utterance is an instantiation of a pattern "*player* passes the ball to *player*" and that all players can appear at the contained slots, thus making the appearance of the example utterance more likely. Notably, semantic classes such as *player* can in principle also be modeled in stochastic language models, in particular by applying class-based models, or in syntactically motivated grammars. Recall that we also experimented with syntactically motivated grammars and with class-based models and that neither classes which were auto-induced on the raw text data nor

syntactically motivated grammars yielded promising results. Thus, using weak supervision in the form of perceptual context information appears to be beneficial for detecting semantic classes compared to working with raw text. An interesting point for future work might be to explore whether using semantic groupings induced by our algorithm in a class-based model yields reasonable results, in particular when applied as a back-off model in combination with a semantically motivated recognition grammar. Further, we have also investigated weighting rules for semantically meaningful lexical units, i.e. in this example the probability for the occurrence of players like "pink nine" and "pink seven" can be increased according to their occurrence frequencies, thus making recognizing them more likely. Our results indicate that by weighting semantically meaningful sequences, performance is improved, possibly because more words carrying semantics are recognized correctly, even though words carrying no semantics like "forward" or "backward" might be confused, which, however, may not prevent correct parsing. In general, while in SLU research mainly cascading systems are explored, in line with previous work (Wang et al., 2003; Bayer and Riccardi, 2012), our results indicate that joint models yield improved parsing performance, even though word recognition performance may decrease. Yet, our results indicate that a combination of a semantic grammar with a standard trigram model during speech recognition can also reduce the word error rate in some cases compared to applying the trigram model only. Furthermore, the results emphasize that capturing semantic information in a language model applied during ASR is beneficial for subsequent semantic parsing, since the ASR can be tuned towards recognizing words carrying semantics more precisely, which is important with respect to parsing performance.

6 Conclusion

This work investigated the induction of semantic grammars applicable for both speech recognition and understanding in a weakly supervised setting, i.e. using ambiguous context information. In doing so, we compared parsing the output of speech recognizers applied with different language models. Our results indicate that by applying the same semantically motivated grammar learned with weak supervision for both recognition and parsing, speech can be parsed into formal meaning representations with a rather low loss in performance compared to parsing of data without recognition errors, that is, textual data or manual transcriptions of speech. An improvement in parsing performance was obtained over a cascading approach in which a standard n-gram model is used, and we have shown how learning weights for grammatical rules applied in speech recognition can yield improved subsequent parsing results compared to applying unweighted grammars.

Acknowledgments

This work has been funded by the DFG within the CRC 673 and the Cognitive Interaction Technology Excellence Center.

References

Ali Orkan Bayer and Giuseppe Riccardi. 2012. Joint language models for automatic speech recognition and understanding. In *Proceedings of the IEEE/ACL Workshop on Spoken Language Technology (SLT)*.

Benjamin Börschinger, Bevan K. Jones, and Mark Johnson. 2011. Reducing grounded learning tasks to grammatical inference. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

David L. Chen and Raymond J. Mooney. 2008. Learning to sportscast: A test of grounded language acquisition. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*.

David L. Chen, Joohyun Kim, and Raymond J. Mooney. 2010. Training a multilingual sportscaster: Using perceptual context to learn language. *Journal of Artificial Intelligence Research*, 37(1):397–435.

Renato De Mori, 2011. *History of Knowledge and Processes for Spoken Language Understanding*, pages 11–40. John Wiley & Sons.

Anoop Deoras, Gokhan Tur, Ruhi Sarikaya, and Dilek Hakkani-Tur. 2013. Joint discriminative decoding of words and semantic tags for spoken language understanding. *IEEE Transactions on Audio, Speech and Language Processing*.

Marco Dinarelli, Alessandro Moschitti, and Giuseppe Riccardi. 2012. Discriminative reranking for spoken language understanding. *IEEE Transactions on Audio Speech and Language Processing*, 20(2):526–539.

Jonathan Fiscus, John Garofolo, Mark Przybocki, William Fisher, and David Pallett. 1998. 1997 english broadcast news speech (hub4). Linguistic Data Consortium.

Judith Gaspers and Philipp Cimiano. 2014. Learning a semantic parser from spoken utterances. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Dan Goldwasser, Roi Reichart, James Clarke, and Dan Roth. 2011. Confidence driven unsupervised semantic parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yulan He and Steve Young. 2005. Semantic processing using the hidden vector state model. *Computer Speech and Language*, 19:85–106.

Ioannis Klasinas, Alexandros Potamianos, Elias Iosif, Spiros Georgiladakis, and Gianluca Mameli. 2013. Web data harvesting for speech understanding grammar induction. In *Proceedings INTERSPEECH*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Helen M. Meng and Kai-Chung Siu. 2002. Semi-automatic acquisition of semantic structures for understanding domain-specific natural language queries. *IEEE Trans. Knowl. Data Eng.*, 14(1):172–181.

Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kai-chung Siu and Helen M. Meng. 1999. Semi-automatic acquisition of domain-specific semantic structures. In *Proceedings EUROSPEECH*.

Zach Solan, David Horn, Eytan Ruppin, and Shimon Edelman. 2005. Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences*, 102(33):11629–11634.

Andreas Stolcke. 2002. SRILM - An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 901–904.

Menno van Zaanen and Pieter Adriaans. 2001. Alignment-Based Learning versus EMILE: A Comparison. In *Proceedings of the Belgian-Dutch Conference on Artificial Intelligence*.

Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. 2004. Sphinx-4: A flexible open source framework for speech recognition. Technical report, Sun Microsystems.

Ye-Yi Wang and Alex Acero. 2003. Combination of CFG and N-gram Modeling in Semantic Grammar Learning. In *Proceedings EUROSPEECH*.

Ye-Yi Wang and Alex Acero. 2005. Sgstudio: Rapid semantic grammar development for spoken language understanding. In *Proceedings of the European Conference on Speech Communication and Technology*.

Ye-Yi Wang and Alex Acero. 2006a. Discriminative models for spoken language understanding. In *Proceedings of the International Conference on Spoken Language Processing*.

Ye-Yi Wang and Alex Acero. 2006b. Rapid development of spoken language understanding grammars. *Speech Communication*, 48 (3-4):390–416.

Ye-Yi Wang, Alex Acero, and Ciprian Chelba. 2003. Is word error rate a good indicator for spoken language understanding accuracy. In *IEEE Workshop on Automatic Speech Recognition and Understanding*.

Ye-Yi Wang, Li Deng, and Alex Acero, 2011. *Semantic Frame-based Spoken Language Understanding*, pages 41–92. John Wiley & Sons.

Chin-Chung Wong and Helen Meng. 2001. Improvements on a semi-automatic grammar induction framework. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*.

Yuk Wah Wong and Raymond J. Mooney. 2006. Learning for semantic parsing with statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Wei-Lin Wu, Ru-Zhan Lu, Jian-Yong Duan, Hui Liu, Feng Gao, and Yu-Quan Chen. 2010. Spoken language understanding using weakly supervised learning. *Computer*, 24:358–382.

Luke S. Zettlemoyer and Michael Collins. 2007. Online Learning of Relaxed CCG Grammars for Parsing to Logical Form. In *Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP)*.