LEARNING A SEMANTIC PARSER FROM SPOKEN UTTERANCES

Judith Gaspers and Philipp Cimiano

Semantic Computing Group, CITEC, Bielefeld University
{jgaspers|cimiano}@cit-ec.uni-bielefeld.de

ABSTRACT

Semantic parsers map natural language input into semantic representations. In this paper, we present an approach that learns a semantic parser in the form of a lexicon and an inventory of syntactic patterns from ambiguous training data which is applicable to spoken utterances. We only assume the availability of a task-independent phoneme recognizer, making it easy to adapt to other tasks and yielding no a priori restriction concerning the vocabulary that the parser can process. In spite of these low requirements, we show that our approach can be successfully applied to both spoken and written data.

Index Terms— Spoken Language Understanding, Semantic Parsing, Lexical Acquisition, Syntactic Acquisition

1. INTRODUCTION

Semantic parsers transform natural language (*NL*) utterances into formal meaning representations (*MR*) and are typically learned from examples consisting of *NL*s annotated with their correct *MR* (e.g., [1][2]). Because such annotations are time-consuming and costly to produce, research has also focused on learning parsers using ambiguous context representations instead of annotations (e.g., [3][4][5][6]) as a step towards building machines which can learn language – analogous to children – through exposure to language in some environment [5]. These parsers are, however, learned from written input. While a word-based automatic speech recognizer (ASR) may be applied in order to handle spoken utterances as typically done in spoken dialogue systems, in this paper we explore how a semantic parser applicable to spoken utterances can be learned directly from spoken utterances coupled with ambiguous semantic annotation without assuming any pre-defined linguistic knowledge bases other than a task-independent phoneme recognizer. Besides the low computational costs for training, this makes it easy to adapt to novel tasks and allows the acquisition of a potentially unrestricted vocabulary by the parser. Furthermore, during parsing the meaning of an utterance can be determined at the whole-sentence level without an a priori restriction regarding possible words – and thus meanings – by the ASR. Yet, as a segmentation task must be tackled additionally, learning a parser is much more challenging when compared to learning from text. Our system performs the segmentation on the basis of unsegmented data, noise and semantic ambiguity by inducing alignments between *NL*s and ambiguous context representations. A parser, represented in the form of a lexicon and an inventory containing syntactic constructions, is then estimated based on co-occurrence frequencies, which are often utilized to establish mappings between form – typically words – and meaning (e.g., [7][8][9]). Alignments are computed both bottom-up by first determining structures of rather low complexity and top-down by including syntactic information; learning linguistic structures of rather low complexity from speech has been addressed previously, e.g., learning (novel) words (e.g., [10][8][11]) or semantically meaningful sequences, so-called *acoustic morphemes* (e.g., [12][13][14]).

2. LEARNING PROBLEM

The input to our system consists of two (temporally paired) channels: a speech channel and a channel with information about the visual context that the learner observes. In particular, the input consists of a set of spoken language utterances, each coupled with a symbolic description of the semantic/visual context by way of predicate logic. All utterances are transcribed by a phoneme recognizer, yielding the input utterances (*NL*) to the learning algorithm. Each *NL* is coupled with a set of actions describing the visual context (*MR*). Each action $mr_i \in$ *MR* consists of a predicate ξ along with a list of arguments $\arg_1, \ldots, \arg_n$, and *NL* corresponds to at most one of the actions. However, direct correspondences are not given, but must be learned by the system instead.[1] We define the underlying vocabulary of the *MR* portion of the data V_{MR} as containing all semantic entities – actions, actors, etc. – that a learner observes visually, i.e. all ξ and arg_i that occur. We define the vocabulary of the *NL* portion of the data V_{NL} as containing all observed phoneme sequences of length 5 to 13. While this may be rather arbitrary, it reduces computational costs, and we assume that sequences of such length already

This work has been funded by the DFG as part of the CRC 673 *Alignment in Communication* and as as part of the Excellence Initiative (CITEC). This work was partially funded within the EU project PortDial (FP7-296170).

[1] In what follows we use examples from the Robocup dataset [5] to illustrate the problem (see Section 4 for details).

cover most "good" candidates for acoustic morphemes.
Given a set of input examples, the goal is to estimate a parser P in the form of a lexicon V_P and an inventory of syntactic constructions C_P, both comprising a meaning for each entry (cf. form-meaning pairings (constructions) [15]). The lexicon V_P consists of acoustic morphemes $a_i \in V_{NL}$ along with their mapping to semantic referents. Each syntactic construction in C_P comprises a syntactic pattern which can contain variable elements (slots), i.e. positions where a $v \in V_P$ may be inserted. The meaning is represented by exactly one semantic frame. If the syntactic pattern contains variable elements, the argument slots in the semantic frame are associated with them by a one-to-one mapping ϕ. An example of an input pair is given by Example 1; desired entries for V_P are "p r= p l EI t" $\rightarrow purple8$ and "p r= p l s @ m @ n t" $\rightarrow purple7$, while the desired syntactic construction is presented in Example 2.

(1)

NL:	p r= p l EI t k I k s t @ p r= p l s @ m @ n t
mr_1:	$playmode(play\_on)$
mr_2:	$pass(purple8, purple7)$
mr_3:	$pass(purple2, purple5)$

(2)

Syntactic pattern	X_1 k I k s t @ X_2
Semantic frame	$pass(ARG_1, ARG_2)$
Mapping (ϕ)	$X_1 \rightarrow ARG_1, X_2 \rightarrow ARG_2$

3. ALGORITHM

The algorithm's work flow is illustrated in Fig. 1. It is roughly divided into four steps: 1) Acquisition of an initial lexicon, 2) bottom-up computation of alignments using the initial lexicon, 3) estimation of a parser based on co-occurrence statistics, and 4) top-down re-estimation of alignments using the learned parser, i.e. lexicon and syntactic patterns, and re-estimation of the parser. Steps 3 and 4 are then repeated until some criterion is met. In order to restrict possible segmentations and computational costs, we apply an unsupervised algorithm, i.e. Bootstrap Voting Experts (BVE)[16][2], to pre-segment all *NL*s into (sub)word-like units. An alignment is estimated given a pre-segmented (*NL*,*MR*) pair by measuring possible segmentations for *NL* along with a hypothesized mapping to semantics for each $mr_i \in$ *MR*. For instance, the desired alignment for Example 1 is presented in Example 3.

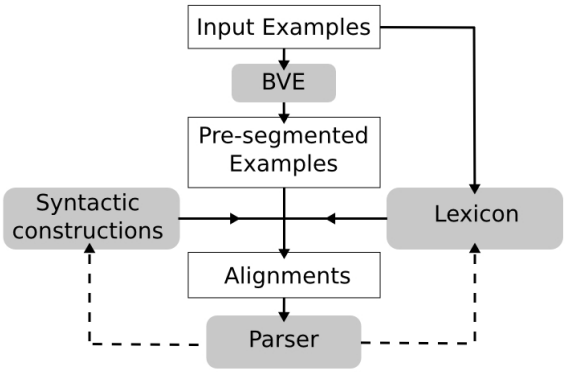

Fig. 1. Work flow of the algorithm.

(3)

NL	X_1 k I k s t @ X_2
mr	$pass(ARG_1, ARG_2)$ $\phi : X_1 \rightarrow ARG_1, X_2 \rightarrow ARG_2$
$nl \rightarrow ref$	p r= p l EI t $\rightarrow purple8$ p r= p l s @ m @ n t $\rightarrow purple7$

Given a list of alignments, a parser is estimated by computing co-occurrence statistics. In particular, we compute association scores at three levels:

1. Lexical L: $nl \rightarrow ref$: between all $v_{nl} \in V_{NL}$ (L_{NL}, e.g. "p r= p l EI t") and $v_{mr} \in V_{MR}$ (L_{MR}, e.g. $purple8$) appearing in alignments.
2. Pattern P: *NL* $\rightarrow mr$: between all patterns (P_{NL}, e.g. "X_1 k I k s t @ X_2") and semantic frames (P_{MR}, e.g. $pass(ARG_1, ARG_2)$) appearing in alignments.
3. Mapping M: ϕ: between all variable positions (M_{NL}, e.g. X_1) and argument slots (M_{MR}, e.g. ARG_1) for each pattern and semantic frame.

Then, $nl \rightarrow ref$ yields V_P, while *NL* $\rightarrow mr$, each coupled with its individual ϕ, yields C_P. The association score $assoc(z_{nl}, z_{mr})$ between a $z_{nl} \in Z_{NL}$ and a $z_{mr} \in Z_{MR}$, $Z \in \{L, P, M\}$ is computed as follows: Let $freq(z_y)$ be the number of observations z_y appears in (at least once). Then

$$assoc(z_{nl}, z_{mr}) = P(z_{nl}|z_{mr}) \times P(z_{mr}|z_{nl}), \quad (4)$$

$$P(z_{nl}|z_{mr}) = \frac{freq(z_{nl}, z_{mr})}{freq(z_{mr})}, P(z_{mr}|z_{nl}) = \frac{freq(z_{nl}, v_{mr})}{freq(z_{nl})}.$$

A $z_{mr} \in Z_{MR}$ is said to be a *meaning* of $z_{nl} \in Z_{NL}$ and z_{nl} *expresses* z_{mr} if $P(z_{nl}|z_{mr}) = \underset{z_i \in Z_{MR}}{\text{argmax}}\ assoc(z_{nl}, z_i)$.
Due to different pronunciations and recognition errors, an algorithm for approximate matching is needed in order to map different phoneme sequences onto each other. We compute the similarity between phoneme strings following Yu et al. (see [17][8]) by first transforming phonemes into vectors of (articulatory) distinctive features and subsequently determining the similarity between two strings based on a modification of the Dynamic Time Warping (DTW) algorithm, where a positive reward is given to matching phonemes and negative scores are assigned otherwise, depending on the number of differing features. In the following, we call the phonetic similarity between two phoneme strings sp_1 and sp_2 $sim(sp_1, sp_2)$, and only strings having at least a certain number of phonemes in common are considered as (potentially) similar, i.e. we set a threshold by multiplying the maximal sequence length with a fraction of the reward set for matching phonemes. In the following, the four learning steps of the algorithm will be explained in more detail.

[2]We utilized the Java implementation available online at http://code.google.com/p/voting-experts/ with parameter optimization via minimum description length.

3.1. Acquisition of an initial lexicon

We assume that at least some sequences appear frequently enough to establish form-meaning mappings. Thus, we compute association scores between all $v_{nl} \in V_{NL}$ and $v_{mr} \in V_{MR}$. For each semantic referent $v_{mr} \in V_{MR}$, we then select a number of sequences having highest association score(s) with v_{mr} as acoustic morphemes for the initial lexicon.

3.2. Bottom-up creation of alignments

Given an example (*NL,MR*), an alignment is created and scored for each $mr_i \in$ *MR*. The parser is then only trained on alignments with maximal score. Given an (*NL,mr*) pair, possible alignments are created by segmenting *NL* such that segments express semantic referents observed in mr according to the initial lexicon. The meaning of a segment s is computed as the meaning mr_s of an entry which has maximal similarity score with s (if existent) $e_s^{L_i}$ in the initial lexicon L_i. The alignment score between s and mr_s is computed as

$$align^{L_i}(s, mr_s) = \frac{sim(s, e_s^{L_i})}{MAXSIM_{mr_s}} * assoc(e_s^{L_i}, mr_s), \quad (5)$$

where $MAXSIM_{mr_s}$ is the maximal similarity which has been obtained for any segment s_i with meaning mr_s and lexicon entry $e_{s_i}^{L_i}$ in one of the segmentations inspected for (*NL,mr*). Thus, an alignment is measured by inspecting whether i) a sequence (e.g., "p r= p l s @ m @ n t") likely corresponds to a lexicon entry (e.g., "p r= p l s E v @ n") and ii) whether this entry is a good expression for an observed argument (e.g., $purple7$). The alignment score for a complete alignment $align$(*NL,mr*) is then computed as the sum of the alignment scores for segments expressing the arguments, i.e.

$$align_{arg}^{L_i}(NL,mr) = \sum_{arg \in ARGs(mr)} align^{L_i}(s, arg). \quad (6)$$

Notice that only those segmentations are considered in which all arguments in mr are indeed expressed by individual segments. Hence, lexical knowledge is also utilized to directly rule out *mr*s which the *NL* does not correspond to.

3.3. Creating a parser

As described previously, given a list of alignments, association scores are computed at the three levels (Lexical, Pattern, Mapping) as defined by equation 4.

3.4. Top-down creation of alignments

If a sequence co-occurs with a referent n-times, then all of its subsequences do so at least n-times and may thus yield better candidates for acoustic morphemes and subsequent segmentation errors. For instance, a sequence "p r= p l I l @ v @ n k I k s" might be incorrectly segmented as "p r= p l X_1 k I k s" because "I l @ v @ n" is a better expression for $purple11$ than "p r= p l I l @ v @ n". We thus apply a top-down step in order to refine alignments based on syntactic knowledge, e.g., once the system has learned that "X_1 k I k s" is a likely expression for $kick(ARG_1)$ while "p r= p l X_1 k I k s" is not, it can use this information to correct the errors described previously. In this step, alignments are computed as in step 2 but in addition a score for segments expressing the predicate is added. Thus, in addition to a lexicon containing acoustic morphemes, a lexicon containing syntactic constructions is utilized; both are extracted from the parser. As in case of creating the initial lexicon, they are created by taking a number of "good candidates" according to the association score, i.e., a number of acoustic morphemes and syntactic constructions are selected for all semantic referents referring to arguments and predicates/semantic frames and stored in lexicon L_a and L_p, respectively. Given an (*NL,mr*), the alignment score is computed as defined in equation 6 as $align_{arg}^{L_a}$(*NL,mr*). The score for segments sp instantiating the pattern $align^{L_p}(sp, mr)$ is computed as defined in equation 5 if the predicate of the lexicon entry for sp matches the observed predicate and summed up with $align_{arg}^{L_a}$(*NL,mr*). The parser is then induced again on the re-estimated alignments (step 3). Steps 3 and 4 are then repeated as long as the cumulative alignment scores increase.

3.5. Parsing

An *NL* is parsed by finding a pattern $p \in C_P$ with acoustic morphemes $a_i \in V_P$ at variable positions for which the sum of the similarity scores with the acoustic morpheme and pattern entries is maximal. If no such match exists, *NL* cannot be parsed. Otherwise, the meaning is the semantic frame associated with p in which the meanings of acoustic morphemes at variable positions are retrieved from V_P and inserted into argument slots according to the mapping ϕ.

4. EVALUATION

The task of learning a semantic parser using ambiguous context information has been previously investigated with respect to learning from written text, not speech, mainly on the RoboCup soccer corpus [5]. In order to compare the employed learning mechanisms to the state-of-the-art, we therefore first evaluate our system on written text, and subsequently present results with respect to application to speech.

4.1. Application to written text

We use the RoboCup soccer corpus [5] for evaluation, which contains four RoboCup games. Game events are represented by predicate logic formulas, yielding the *mr*s. The games were commented by humans, yielding the written *NL* utterances. For example, *pass(purple10,purple7)* represents an *mr* for a passing event which might be commented as "purple10

kicks to purple7". In the corpus, each *NL* comment is paired with a set of possible $mr_i \in MR$. These correspond to all actions observed five or less seconds prior to the comment. The data is ambiguous in that it is unknown which of the actions in the set – if any – is described by the utterance. The corpus also contains a gold standard comprising *NL*s annotated with their correct *mr*s. We evaluated our approach in line with Chen et al. [3] by performing 4-fold cross-validation. Training was done on the ambiguous training data, while the gold standard for a fourth game was used for testing. Results are 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 [3].

To our knowledge, the best performing system so far has been proposed by Börschinger et al. [4], who tackled the task by inducing a Probabilistic Context Free Grammar, achieving an F_1 of 86%. When applied to text, using our proposed learning mechanisms a parser can be build straightforwardly. We computed an initial lexicon by taking all uni- and bigrams as the vocabulary V_{NL} and computed alignments only once by applying a single bottom-up step. Approximate matching, while not needed when computing alignments, was applied during parsing of *NL*s for which no pattern could be found otherwise. In that case the system searched for i) a match with a Levenshtein distance of 1, (e.g. "Pink1 makes a cross pass" can be matched with "X_1 makes a pass") and subsequently ii) a partial match (e.g., "Pink1 passes to Pink2 near midfield" can be matched with "X_1 passes to "X_2"). Results are presented in Table 1, indicating that when applied to written text the algorithm yields state-of-the-art performance, slightly outperforming Börschinger et al. [4].

Table 1. Results

Input	**Parser**	F_1 **(%)**
Written text	Börschinger et al. (2011)	86.0
Written text	Our system	88.7
Grapheme-to-phoneme	Baseline	18.9
Grapheme-to-phoneme	Our system	82.8
ASR phoneme	Baseline	0.3
ASR phoneme	Our system minus top-down	58.0
ASR phoneme	Our system incl. top-down	64.2

4.2. Application to speech

Allowing to explore learning from spoken utterances, all *NL* utterances in the RoboCup training data were read by a native American speaker. Out of these examples, 23 were excluded due to an error made by the speaker, yielding 1849 examples. All spoken utterances were then transcribed using a phoneme recognizer.[3] Furthermore, we applied grapheme-to-phoneme conversion to the written text using MaryTTS [19] for evaluating the algorithm without recognition errors. By comparing the ASR transcribed to the converted data, a phoneme error rate (PER) of 34.2% averaged over all games was obtained.

In order to evaluate the amount of language learned by our system, as a baseline we computed the F_1 score that would be achieved if the system would have performed "rote learning" of input examples. In particular, an *NL* in the test data was parsed – if it had also been observed in the training data – by choosing one of the *mr*s observed with it randomly.[4]

Results are presented in Table 1. In case of unsegmented phoneme sequences without recognition errors (grapheme-to-morpheme) still a high F_1 of 82.8% is obtained, indicating that the proposed segmentation mechanisms are appropriate.[5] It must be noted that while expectedly performance degrades when working with the ASR transcriptions, this would be the case – at least to some extent – when applying a word-based ASR as well. Yet, the results are promising, showing that in spite of noise and contextual ambiguity, it is still possible to learn a parser which can be utilized to understand several unseen utterances, as indicated by the large increase in F_1 when compared to the baseline. It must be noted that the RoboCup corpus may be complicated in that several sequences expressing referents have subsequences in common, i.e. most expressions for players start with either the prefix "purple" or "pink" followed by a number. Thus, what mainly distinguishes referents are the numbers, and due to recognition errors sometimes only subsequences expressing numbers were associated, yielding segmentation and parsing errors because the prefix is needed for determining the correct referent. In case of ASR output, applying the top-down step additionally yielded an improvement over applying only the bottom-up step, indicating that syntactic knowledge – which is typically ignored in algorithms for unsupervised segmentation – can indeed provide useful segmentation cues, at least when working with noisy sequences.

5. CONCLUSION

This paper presented a method for learning a semantic parser applicable to spoken utterances. We have shown that the presented learning mechanisms yield a parser achieving state-of-the-art performance in case of textual input. Furthermore, our results indicate that even in spite of noise and contextual ambiguity, in case of spoken utterances it is still possible to learn a parser which can be used to parse unseen spoken utterances.

[3] We applied sphinx-3 [18] with the configuration and resources available online at http://cmusphinx.sourceforge.net/wiki/phonemerecognition. Silence was removed from the transcriptions; transcriptions were converted from ARPABET into X-SAMPA, allowing comparison to MaryTTS output.

[4] Notice that in case of ASR output the baseline is very low as due to recognition errors it is the case that only a single *NL* appears in both the training and the test data for two folds, in one case together with 8 and in the other case together with 3 possible *mr*s. By applying approximate matching the baseline can be increased to F_1 = 19.6%.

[5] Notice that in this case expressions for referents can also be found by coupling co-occurrence frequencies with a length bias [20].

6. REFERENCES

[1] Yuk Wah Wong and Raymond J. Mooney, "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*, 2006.

[2] Luke S. Zettlemoyer and Michael Collins, "Online learning of relaxed ccg grammars for parsing to logical form," in *Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007.

[3] David L. Chen, Joohyun Kim, and Raymond J. Mooney, "Training a multilingual sportscaster: Using perceptual context to learn language," *Journal of Artificial Intelligence Research*, vol. 37, no. 1, pp. 397–435, 2010.

[4] Benjamin Börschinger, Bevan K. Jones, and Mark Johnson, "Reducing grounded learning tasks to grammatical inference," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011.

[5] David L. Chen and Raymond J. Mooney, "Learning to sportscast: A test of grounded language acquisition," in *Proceedings of the 25th International Conference on Machine Learning*, 2008.

[6] Judith Gaspers, Philipp Cimiano, Sascha Griffiths, and Britta Wrede, "An unsupervised algorithm for the induction of constructions," in *Proceedings of the joint IEEE International Conference on Development and Learning and on Epigenetic Robotics*, 2011.

[7] Afsaneh Fazly, Afra Alishahi, and Suzanne Stevenson, "A probabilistic computational model of cross-situational word learning," *Cognitive Science*, vol. 34, no. 6, pp. 1017–1063, 2010.

[8] Chen Yu, Dana H. Ballard, and Richard N. Aslin, "The role of embodied intention in early lexical acquisition," *Cognitive Science*, vol. 29, pp. 961–1005, 2005.

[9] Judith Gaspers and Philipp Cimiano, "A computational model for the item-based induction of construction networks," *Cognitive Science*, in press.

[10] Deb Roy and Alex Pentland, "Learning words from sights and sounds: A computational model," *Cognitive Science*, vol. 26, no. 1, pp. 113–146, 2002.

[11] Ryo Taguchi, Naoto Iwahashi, Takashi Nose, Kotaro Funakoshi, and Mikio Nakano, "Learning lexicons from spoken utterances based on statistical model selection," in *Proceedings Interspeech*, 2009.

[12] A. L. Gorin, D. Petrovska-Delacrétaz, G. Riccardi, and J.H. Wright, "Learning spoken language without transcriptions," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 1999.

[13] Michael Levit, Elmar Nöth, and Allen Gorin, "Using em-trained string-edit distances for approximate matching of acoustic morphemes," in *Proceeding Interspeech*, 2002.

[14] Christophe Cerisara, "Automatic discovery of topics and acoustic morphemes from speech," *Computer Speech and Language*, vol. 23, no. 2, pp. 220–239, 2009.

[15] Adele Goldberg and Laura Suttle, "Construction grammar," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 1, no. 4, pp. 468–477, 2010.

[16] Daniel Hewlett and Paul Cohen, "Bootstrap voting experts," in *Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence*, 2009.

[17] Chen Yu and Dana H. Ballard, "A computational model of embodied language learning," Tech. Rep., Department of Computer Science, University of Rochester, 2002.

[18] P. Placeway, S. Chen, M. Eskenazi, U. Jain, V. Parikh, B. Raj, M. Ravishankar, R. Rosenfeld, K. Seymore, M. Siegler, R. Stern, and E. Thayer, "The 1996 hub-4 sphinx-3 system," in *Proceedings of the DARPA Speech recognition workshop*, 1997.

[19] M. Schröder and J. Trouvain, "The german text-to-speech synthesis system mary: A tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, pp. 365–377, 2003.

[20] Judith Gaspers and Philipp Cimiano, "A usage-based model for the online induction of constructions," in *Proceedings of the joint IEEE International Conference on Development and Learning and on Epigenetic Robotics*, 2012.