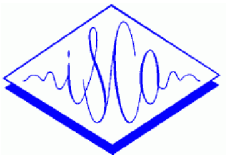
ISCA Archive
http://www.isca-speech.org/archive



INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

A Data-driven Approach to Understanding Spoken Route Directions in Human-Robot Dialogue

Raveesh Meena, Gabriel Skantze, Joakim Gustafson

Department of Speech Music and Hearing, KTH, Stockholm, Sweden
raveesh@csc.kth.se, gabriel@speech.kth.se, jocke@speech.kth.se

Abstract

In this paper, we present a data-driven chunking parser for automatic interpretation of spoken route directions into a route graph that is useful for robot navigation. Different sets of features and machine learning algorithms are explored. The results indicate that our approach is robust to speech recognition errors.

Index Terms: spoken language understanding, route directions, human-robot interaction

1. Introduction

Robots are increasingly finding place in our daily lives. This transition from constrained and well-controlled industrial settings into dynamic environments where the objectives and situations may change radically over time makes it infeasible to equip robots with all necessary knowledge a-priori. While robots can learn from experience, understanding something hitherto unknown remains a challenging task for them. Humans, however, are a rich source of information. By engaging in a spoken dialogue with humans robots can extract this information and gain knowledge about the unknown.

How robots can be endowed with skills for spoken dialogue with humans and seek route directions to navigate their way in unknown real urban environments is the central research question that is being investigated in the IURO project[1]. A common way of representing navigational knowledge is the *route graph*. In a previous study [1] we presented a novel approach for data-driven semantic interpretation of manually transcribed route instructions (in Swedish) into route graphs. The results indicated that it is possible to get people to freely describe routes which can be automatically interpreted into a route graph. In this paper, we discuss and present the findings of three extensions to our previous work. First, we now also learn the route segments of a route graph. Second, we evaluate the usability of our approach on route instructions given in English. Third, we evaluate our approach on automatic speech recognition (ASR) results of spoken route instructions.

2. Motivation and related work

2.1. Data-driven semantic interpretation

The problem of interpreting spoken route instructions into a route graph is that of semantic interpretation of spoken utterances – or Spoken Language Understanding (SLU), as it is commonly referred to – in dialogue system processing. The problem of SLU can be formulated as taking a speech recognition result and producing a semantic representation that can be used by the dialogue manager to decide what to do next. Automatic speech recognition is, however, prone to errors and poses challenges for SLU. The real world setting of the IURO robot makes this even more difficult.

In a study [2] on human-human dialogue with an error prone speech recognition channel it has been shown that humans that have a very clear goal of the interaction may accurately pick out pieces of information (despite poor accuracy in speech recognition) from the speech recognition results that are relevant to the task, and ask relevant task related questions to recover from the problem. This may suggest that a data-driven approach to *keyword spotting* where specific words in the input are associated with certain concepts or slot-value pairs might server our purpose. However, the semantics of route directions is highly structured and cannot be treated as "bag of words/concepts". For example, the route instruction "*at the roundabout take the second exit eh on your left hand side*" contains not just the concepts ROUNDABOUT, TAKE, EXIT and LEFT, but also the relationship between the concepts. That the action of changing direction should be taken at the roundabout and the direction to take is left and it is precisely the second exit and not any other. Any approach to automatically interpret this route instruction must preserve this structural relationship.

Another demand for understanding freely spoken route directions is that the data-driven approach should be able to generalize to some extent when encountered with unseen concepts, and be able to back off to more general concepts, without breaking the conceptual structure. We therefore need a domain model (ontology) which defines concepts on different levels of specificity and specifies how they may be structured and the data-driven approach should take this domain model into account.

An approach to SLU using Markov Logic Networks is presented in [3]. However, the resulting semantic representations are limited to a set of slot-value pairs (i.e., they are not structured). A Hidden Vector State model based approach in [4] is shown to learn deeply structured semantic interpretations in a travel-booking domain. However, it is not clear whether the approach may utilize an ontology and back off to more general concepts in order to learn generalizations. In another approach to SLU a context-free grammar (CFG) is augmented with semantic instructions [5]. However, the approach assumes that the input may be described with a CFG which makes it unsuitable for interpreting freely spoken route directions. In our previous work [1] we presented a data-driven approach that is able to interpret and represent the structural relations present in route directions, and learns generalization using a domain ontology. However, the problem of interpreting spoken route instructions was left as future task which we have addressed in our current work.

[1] Interactive Urban Robot (www.iuro-project.eu)

2.2. Route graphs for navigation

A *conceptual route graph* (CRG) is a type of a route graph that represents the semantics of human route descriptions. In a scheme proposed in [6], the nodes in a CRG represent places where a change in direction takes place and edges connect these places. The CRG may be divided into *route segments*, where each segment consists of an edge and an ending node where an action to change direction takes place. Conceptually, segment consists of (i) *controllers* – a set of descriptions that guide the traversal along the edge, e.g., "go straight down", (ii) *routers* – a set of place descriptors that helps to identify the ending node, and (iii) *action* – the action to take at the ending node in order to change direction. At least one of these three components is required in a route segment.

Figure 1 illustrates a CRG. The nodes in the graph represent the concepts and the edges their attributes. The concepts, their attributes and argument types are defined in the type hierarchy of the domain model using the specifications in the dialogue framework Jindigo [7].

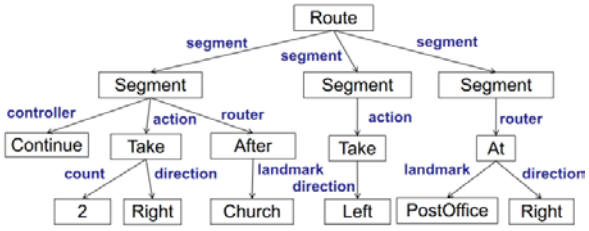


Figure 1: *The conceptual route graph for the route instruction "go straight and take the second right after the church then eh take a left the post is on the right hand side"*

3. A Chunking parser for semantic interpretation

In our previous work we presented a novel application of Abney's *chunking parser* [8] to data-driven semantic interpretation. Inspired by the *Chunker* and the *Attacher* stages of the syntactic analyzer, we used the Chunker for finding base concepts in a given sequence of words. The Attacher is then given the task of assigning more specific concepts (given by the type hierarchy of the domain) and to attach concepts as arguments. For example, route instruction "*turn right at the post office and eh continue till the church*", could be chunked as the following:

[ACTION turn] [DIRECTION right] [ROUTER at] [LANDMARK the post office] [DM and] [FP eh] [CONTROLLER continue till] [LANDMARK the church]

To turn chunking into a classification problem, we followed the common practice of assigning two labels for each type of chunk: one with prefix B- for the first word in the chunk and one with prefix I- for the following words. The Attacher takes a base concept (a chunk) and does two things: First, it may assign a more specific concept class (like CHURCH). To allow it to generalize, the Attacher also assigns all ancestor classes, based on the domain model (i.e., BUILDING for CHURCH; this, however, is not shown in the example). The second task for the Attacher is to assign attributes and assign them values. Some attributes are filled with new concepts (like *property*: LARGE), while others are attached to the nearest concept that fits the argument type according to the domain model (like *direction*: →, which means that the interpreter should look for a matching argument in the right context). The Chunker output from above could be modified by the Attacher as in the following example:

[TAKE(*direction*: →) turn] [RIGHT right] [AT(*landmark*: →) at] [POSTOFFICE the post office] [DM and] [FP eh] [CONTINUE(*landmark*: →) continue till] [CHURCH the church]

The Chunker in our approach is a *single-label classifier* and the Attacher a *multi-label classifier* where none, one or several labels may be assigned to the chunk. Naive Bayes and a set of Linear Threshold Units algorithms were implemented for the classifiers and tested. Only the latter can be used for the multi-label learning in the Attacher. As a final step, heuristic rules were used to group the CONTROLLERS, ROUTERS and ACTIONS into SEGMENTS (and a ROUTE). The final result of the Chunking parser is basically a conceptual route graph (cf. Figure 1).

In our previous study, we used data collected in a Wizard-of-Oz experiment. Subjects first watched a recorded video of a route and then described (in Swedish) the route direction to the wizard. Cross-validations were performed on a set of 35 route instructions which were manually annotated with the correct chunking and attachments. To measure the performance of the Chunking parser we compared the resulting CRG with its reference CRG (the manual annotation). In analogy to the measure of Word Error Rate (WER) in evaluation of speech recognition results, we used Concept Error Rate (CER) as a measure of edits required in the reference CRG to obtain the generated CRG, by flattening the conceptual trees. A penalty metric was defined to assign weights to each type of edit operation based on the type hierarchy in our domain model. The weighted CER thus obtained was better representative of the true performance of the Chunking parser in contrast to simple edit distance. The best performance (CER of 21.47) for the Chunker was achieved with a Sparse Perceptron learner and for the Attacher (CER of 25.60) using a Sparse Averaged Perceptron learner. These figures indicated that it is possible to get people to freely describe routes which can be automatically interpreted into a conceptual route graph.

4. Method

In the study presented here, we made three extensions to our previous work. First, inspired by the performance of the Chunker in learning base concepts, we introduced another chunk learner – the *Segmenter* – to find *route segments* in the sequence of chunks. The Chunker output shown earlier can be segmented as follows:

[SEGMENT [ACTION turn] [DIRECTION right] [ROUTER at] [LANDMARK the post office] **] [SEGMENT** [DM and] [FP eh] [CONTROLLER continue till] [LANDMARK the church] **]**

The Attacher performs the same tasks as earlier, except that it now looks for attachments only within the current route segment. The Chunking parser output now contains route segments which were not learned in our previous work. This makes the semantic interpretation of route instruction into route graphs fully automatic. Second, we verified whether the Chunking parser can be applied for extracting CRGs from route instructions given in

English. Third, we addressed the problem of extracting CRGs from speech recognition results of spoken route instructions. In order to meet these two objectives we used the Instruction-Based Learning (IBL) corpora [9] of route instructions. The corpus contains manual transcriptions and audio recordings of 144 spoken route instructions given in English. In IBL, 24 subjects were recorded as they gave route instructions to a miniature robot in a miniature town. The town environment comprised of various landmarks from real urban setup and provides a close approximation of the urban settings in IURO. The average length of instructions in IBL is 56 words. The manual transcriptions include filler pauses, self-corrections and repetitions. The audio recordings were made in a normal indoor setup, and contain speaker pauses and silences during route direction giving.

As a first step, the Chunking parser's performance was evaluated on the manual transcriptions to obtain the baseline for comparing its relative performance on the speech recognition results. A set of 30 IBL route instructions were manually annotated. This data, on an average, contained 31.1 words, 13.13 concepts and 2 route segments per route instruction and was used as cross-validation set.

Next, we trained the language model of an off-the-shelf ASR system with 113 route instructions (excluding those in the cross-validation set) from the IBL corpora. The trained ASR had a vocabulary size of 213 words. The audio recordings of the route instructions in the cross-validation set were recognized by the trained ASR. The best hypothesis for each instruction was chosen for validating the Chunking parser's performance.

We tested the performance of the Chunking parser again on the Naive Bayes (NB) and Linear Threshold Units (LTU) algorithms. Two types of LTUs were tested: Sparse Perceptron (SP) and Sparse Averaged Perceptron (SAP) [10]. For the Chunker the following features were used: *word instance:* the word itself, *word window*: one previous and two next words, *previous tags*: chunk tags of the two previous words, and *POS window*: part of speech of the words in the word window. For the Segmenter and the Attacher we used: *bag of words*: an ordered set of words in the chunk, *bag of POS*: an ordered set of POS of words in the chunk, and *chunk label window:* a window of chunk labels of two previous and one next chunk.

5. Results

For drawing comparisons we used the baseline performances of the Chunking parser corresponding to a *keyword spotting* based method (shown in Table 1 in bold figures).

	Features	CER_{NB}	CER_{SP}	CER_{SAP}
Chunker	Word instance	**50.83**	46.15	45.17
Segmenter	Bag of words	**77.22**	60.83	53.06
Attacher	Bag of words	--	77.62	**77.70**

Table 1: *Baseline performances of the Chunking parser.*

In general, the LTUs performed better than Naive Bayes for the chunking task. The best performance for the Chunker is achieved with the Sparse Averaged Perceptron learner, as shown in Table 2, where the CER is shown with additive feature sets for the three algorithms.

For the Segmenter the best performance is achieved with the Sparse Perceptron learner and using the *chunk label window* feature, as shown in Table 3.

Features	CER_{NB}	CER_{SP}	CER_{SAP}
Word instance	50.83	46.15	45.17
+ Word window	17.31	21.33	20.82
+ Previous tags	18.16	10.86	**10.64**
+ POS window	19.61	11.01	11.81

Table 2: *Chunker performances with additive features.*

Features	CER_{NB}	CER_{SP}	CER_{SAP}
Bag of words	83.89	64.17	59.44
+ Bag of POS	98.33	67.50	64.17
Chunk label window	31.67	**25.83**	28.89

Table 3: *Segmenter performances with additive features.*

The best performance for the Attacher was achieved with the Sparsed Averaged Perceptron learner and using the *bag of words* feature alone, as shown in the first two columns of Table 4.

A closer look at the route segments indicated that poor placement of route segment boundaries resulted in a restricted search space for finding attachments. In order to get an estimate of the Attacher's performance independently from the Segmenter, we compared only the sub-graphs in all the route segments of a CRG with their counterparts in the reference CRG. The last column in Table 4 presents the Attacher's performance following this scheme.

Features	CER_{SP}	CER_{SAP}	$sgCER_{SAP}$
Bag of words (BW)	29.42	**29.11**	**19.99**
Chunk label window	49.32	49.74	50.93
+ BW	29.62	29.44	21.75
BW + Bag of POS	30.85	31.48	21.69

Table 4: *Attacher performances with additive features.*

The learning curves in Figure 2 show that while the Chunker, the Segmenter and the Attacher were able to perform well with little training (just 15 samples) their performance continues to improve with more training data.

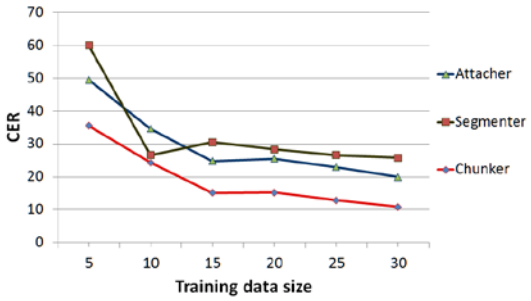


Figure 2: *The learning curves for the Chunking parser.*

The performance of the Chunking parser on the recognized hypotheses of spoken route instructions is illustrated in Figure 3. While the first eight points on the horizontal axis represent the Chunking parser performances w.r.t. various ASR performances (obtained by varying the language model parameters), point nine is the performance on transcribed route instructions, i.e., WER of 0. The CER curves suggest that the performance of the Attacher follows the WER. Against the best ASR performance, WER of 25.02, the Attacher achieved the CER of 40.55. In contrast to the Attacher CER of 19.99 on transcriptions, the introduction of a

WER of 25.02 resulted in a *relative* CER of 15.53. This is comparable with the Attacher's original performance (CER of 19.99). The rather steady relative CER curve (R-CER) in Figure 3 highlights the robustness of our approach in dealing with errors in speech recognition and the irregularities of spoken language. The following example from the validation set highlights the strength of the Chunking parser:

Human: "*okay if you erm if you get to the university and then take the third first exit just past the university and then turn right erm boots is on your left*"
ASR hypothesis: "*go front again the university and then take the third first exit just past the university and then turn right and trees is on your left*",
Chunking parser output:
[CONTINUETO(*landmark*:→) go front again] [UNIVERSITY the university] [DM and then] [TAKE(*count*: →, *landmark*:→) take] [NUMBER the third] [EXIT first exit] [AFTER(*landmark*:→) just past] [UNIVERSITY the university] [DM and then] [TAKE(*direction*:→) turn] [RIGHT right] [DM and] [LANDMARK trees] [AT(*direction*: →,*landmark*: ←) is] [LEFT on your left]

The Chunking output shows that, (i) despite errors in the recognized hypothesis the conceptual information about going to the university present in spoken utterance was identified and represented (CONTINUETO(*landmark*:→)) in the route graph, and (ii) on encountering the unseen concept "*trees*" that doesn't exist in our domain model, the Attacher has backed off and assigned the more general concept LANDMARK to it. In this way, the Attacher has not only preserved the fact that a landmark has been mentioned, but also maintained its structural relationship to the concept AT and LEFT in the route graph. This is where a spoken dialogue could be used to clarify the unknown concept.

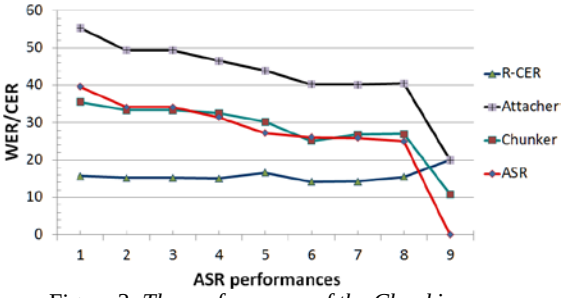


Figure 3: *The performance of the Chunking parser w.r.t ASR WER (R-CER: relative CER).*

6. Conclusions

The encouraging performance scores of the Chunking parser on the English corpus (CER of 19.99 vs. baseline 77.70) and the Swedish corpora (CER 25.60 vs. baseline 83.34) indicate that our data-driven approach could be easily used for understanding route instructions in other languages using simple features. Features such as affixes and part of speech which were found informative for interpreting route instructions in Swedish were not useful for English. This might be explained by the fact that Swedish has a richer morphology in comparison to English.

In contrast to the baseline performance of *keyword spotting* based method for route instruction interpretation (CER of 77.70) the Attacher's performance (CER of 19.99) is very promising. Moreover, the rather steady relative CER on recognized route instructions shows that the presented framework is robust to speech recognition errors. These are encouraging results for using Chunking parser in a spoken dialogue system.

The performance of the automatic Segmenter was not in line with the performance of the Chunker, which means that we still might need to use heuristics for this task. To find better algorithms for automatic segmentation is a topic for future work.

Our next step is to evaluate the usability of the route graphs generated by the Chunking parser. One way to do this is asking human subjects to follow the paths in the IBL miniature town using the conceptual information present in CRGs. How close the subjects reach to the goal could indicate the utility of the route graphs. The information about the key concepts that subjects find useful in navigation may indicate the information they would incorporate when giving directions. This could help in user modeling and for designing dialogue strategies for the IURO robot when seeking route directions from passersby.

The reasonably good performances of the Chunker and the Attacher with little use of right context encourages us to train the Chunking parser on the left context only, and evaluate its performance on the Jindigo framework for incremental dialogue processing [7].

7. Acknowledgements

This work has been carried out at the Centre for Speech Technology at KTH, and is supported by the European Commission project IURO, grant agreement no. 248314. The authors would also like to thank their colleagues for their valuable comments on this work

8. References

[1] Johansson, M., Skantze, G., & Gustafson, J. (2011). Understanding route directions in human-robot dialogue. In *Proceedings of SemDial* (pp. 19-27). Los Angeles, CA.
[2] Skantze, G. (2005). Exploring human error recovery strategies: implications for spoken dialogue systems. *Speech Communication, 45*(3), 325-341.
[3] Meza-Ruiz, I. V., Riedel, S., & Lemon, O. (2008). Accurate statistical spoken language understanding from limited development resources. In *Proceedings of ICASSP 2008* (pp. 5021-5024). Las Vegas, Nevada.
[4] He, Y., & Young, S. (2006). Spoken language understanding using the hidden vector state model. *Speech Communication, 48*(3-4), 262-275.
[5] Wong, Y. W., & Mooney, R. (2007). Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of ACL-07* (pp. 960-967). Prague, Czech Republic.
[6] Müller, R., Röfer, T., Lankenau, A., Musto, A., Stein, K., & Eisenkolb, A. (2000). Coarse qualitative descriptions in robot navigation. In Freksa, C., Brauer, W., Habel, C., & Wender, K-F. (Eds.), *Spatial Cognition II* (pp. 265-276). Springer.
[7] Skantze, G. (2010). *Jindigo: a Java-based Framework for Incremental Dialogue Systems*. Technical Report, KTH, Stockholm, Sweden.
[8] Abney, S. (1991). Parsing by chunks. In Berwick, R. C., Abney, S. P., & Tenny, C. (Eds.), *Principle-Based Parsing: Computation and Psycholinguistics* (pp. 257–278). Dordrecht: Kluwer.
[9] Kyriacou, T., Bugmann, G., & Lauria, S. (2005). Vision-based urban navigation procedures for verbally instructed robots. *Robotics and Autonomous Systems, 51*(1), 69-80.
[10] Collins, M. (2002). Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of ACL* (pp. 1-8). Philadelphia, PA.