Dept. for Speech, Music and Hearing

Quarterly Progress and Status Report

Example based shallow semantic analysis in the August spoken dialogue system

Lindberg, N. and Gustafson, J.

KTH Computer Science and Communication

journal: TMH-QPSR
volume: 41
number: 1
year: 2000
pages: 039-044

http://www.speech.kth.se/qpsr

Example based shallow semantic analysis in the August spoken dialogue system

Nikolaj Lindberg and Joakim Gustafson
Centre for Speech Technology (CTT), Royal Institute of Technology, Stockholm, Sweden
{nikolaj, jocke}@speech.kth.se

Abstract

In this paper, the semantic analyser component of the August spoken dialogue system is presented. The task of the semantic analyser is to transform the output of a speech recogniser into a flat semantic representation, used by the dialogue manager. The analyser produces a non-compositional shallow semantic representation for each of the hypotheses in an N-best list produced by the speech recogniser. The analyser uses a machine-learning system to build its analyses from an example database. An analysis is obtained by running three independent classifiers in parallel, and concatenating the results from the different classifiers. One of the classifiers recognises unlikely utterances, and is used as a negative filter to identify (semantically) implausible hypotheses. Another predicts the topic of an utterance, while the third one returns a flat feature-value representation.

Introduction

This paper presents a shallow semantic analyser that was used in an experimental spoken dialogue system. This system was available for testing in a public place for six months. The semantic analyser, which analysed the output from a speech recogniser, was fully example-based — no semantic or grammatical or rule based processing took place. An utterance hypothesis produced by the speech recogniser was given the analysis of a similar example in an example database. A semantic analysis was obtained by concatenating the results from three independent sub-analyses. The analyser server was built around the freely available TiMBL memory-based machine learning system, developed at Tilburg University (Daelemans et al., 1998).

The approach taken in this work was motivated by the need for rapid development of a semantic interpreter that would be easy to extend. The coverage of the analyser is extended simply by adding more annotated examples to the training database. There were three main fields making up the semantic analysis, each of which was filled out by an independent classifier. The first field stated whether an utterance was acceptable or not (`y` or `n`), the second field predicted the topic of the utterance (e.g., `main`, `meta`, `strindberg`, `stockholm`, `yellow_pages`, ...) and the third field was instantiated with a flat feature-value representation of the utterance (e.g. `{object: restaurant, place:mariatorget}`). The semantic representation was shallow in that it consisted of a relatively simple feature-value structure, and was intended to make adequate distinctions from the dialogue system perspective rather than to constitute a “general” semantic component. During the development of the component, the semantic analyser was itself used in a graphical tool for annotating the training data.

The novel contribution of the work described here is the use of memory based classifiers to rapidly construct a robust semantic analyser to apply to the output of a speech recogniser. No grammar or semantic rules are used, only examples of correctly analysed utterance hypotheses.

The August spoken dialogue system

The Swedish August system (Gustafson et al., 1999) featured an animated agent (named after the author August Strindberg) with which the user interacted (Figure 1). The user communicated with the system by means of voice input only. The animated agent communicated with synthetic speech, facial expressions and head movements. In addition, August had a thought balloon in which text which was not to be synthesised could be displayed (such as help messages or tips — or a remark which August “thought” but did not say aloud). In addition to the screen showing the animated agent, there was a second screen displaying results from database queries (such as street maps indicating where some restaurant was located), etc.

Figure 1. August quotes Strindberg: I am one hell of a man, and I know a couple of tricks.

The August system included the following components:

- Speech synthesis (Carlson et al., 1990)
- Lip-synchronised 3D animated "talking head" with a rich repertoire of facial expressions and head movements (including twisting the moustache) (Beskow, 1997)
- Camera eye which detected user movement (Öhman, 2000)
- Speech recogniser for continuous speech, including a confidence estimation (Ström, 1997)
- Dialogue manager
- General broker architecture for handling the distributed system modules, running under different platforms
- Example based semantic analyser, described below

The idea of handling different domains was an important aspect of the project. The user could ask the system e.g. where to find restaurants and other facilities in Stockholm. Apart from the spoken feedback, the results from database queries could be presented as graphics and text. In addition to the "yellow pages" domain, the user could ask things about Stockholm, August Strindberg and speech technology. Furthermore, the system was able to do some social interacting, in the form of greetings, etc. To encourage people from the public to talk to August, the system ran in a "chat-loop" when no one was using it, explaining things about itself, dropping facts about Stockholm or quoting the works of August Strindberg. The animated agent had a distinctive personality, which often invited the users to start the interaction with a couple of socialising turns (Bell & Gustafson, 1999a).

One of the goals of the August project was to demonstrate how existing speech technology modules could be put together to rapidly prototype a spoken dialogue system. A second important goal was to get out of the laboratory and expose speech technology to a real world public, and the aim was to collect genuine data from a general public from the start of the project. In order to elicit as spontaneous utterances as possible, the demo system was designed without a single strict domain (such as e.g. ticket reservations). The August spoken dialogue system was available in a public setting in downtown Stockholm between September 1998 and March 1999. A database of 10,058 spontaneous user utterances produced by 2,685 different users was collected and transcribed. This user data has been thoroughly analysed, and reported on in e.g. Bell & Gustafson (1999a) and Bell & Gustafson (1999b).

The semantic analyser component

The component described in the present work analysed the output from the speech recogniser, and translated a user utterance into a simple semantic representation, used by the dialogue manager to give a relevant response. The "rapid prototyping" nature of the project put constraints on the semantic analyser: It had to be developed in a short time, and it should be robust and easy to extend. The analyser should also be as independent as possible from the other modules, since these were under development too. These requirements excluded a complex semantic representation, and it also excluded a complex syntactic analysis. Instead, an example-based approach, excluding any grammatical processing, was chosen. It was assumed that it should be easier to construct a database of semantically analysed examples rather than to e.g. hand-craft rules for semantic analysis.

It could be noted that the approach described in this paper is not an instance of key-word spotting, since complete utterance hypotheses are used as input to the semantic component: The developer of the semantic analyser has not explicitly identified any key-words.

The TiMBL machine-learning system

TiMBL (Daelemans et al., 1998) is a freely available memory-based learning system developed at Tilburg University. TiMBL has been used in a number of different natural language processing applications.

TiMBL classifies a new (test) example based on analogy (similarity) between previously seen (training) examples. An example (test as well as training) is represented as a fixed length feature-value vector. In addition to the feature-value vector, a training example has a field that contains the correct classification of the example. A test example is assigned the classification of the most similar training example.

The philosophy of memory-based learning is to keep all examples in the training data — exceptions or unusual examples should be represented as well. In memory-based learning, the learning phase is relatively simple compared to e.g. rule inducing methods, in which bigger efforts have to be made finding generalisations over the training data. However, in order for a tool such as TiMBL to be useful for a given problem, it has to be possible to formulate the problem as a classification task where the examples can be represented as fixed length feature-value vectors.

The TiMBL system, which is efficient and easy to use, implements different ways of representing and searching the example database with which the user might experiment.

Training data

The analyser training data was semi-automatically annotated with the appropriate semantic representation, and consisted of some 2,000 utterance hypotheses produced by a speech recogniser. In the initial stage of the project, a set of possible user utterances, which the system should be able to handle, was created. Six different speakers were recorded producing 60 different utterances each. The recordings were run through the speech recogniser, and the output lists of hypotheses made up the initial example base on which the semantic analyser was trained. Each unique hypothesis produced in this manner was manually analysed. A graphical tool for annotating the corpus was created. The tool suggested an analysis for an utterance, and the annotator either accepted this hypothesis or changed it. The suggestions were produced by the semantic analyser itself, so the first few hundred analyses had to be manually annotated altogether, since no training data existed to bootstrap from.

The training examples that TiMBL was trained on consisted of word sequences, the actual utterance hypotheses in the training data, and sets of semantic tags, obtained from the recogniser lexicon. There was also a feature that indicated the number of lexical items in the hypothesis. All three classifier processes, described below, used the same training data, except the classifier field.

The fixed length format meant that there had to be a maximum length of lexical items in an utterance. This was set to seven, a number chosen after inspecting the training data. Since the lexicon included many multi-word sequences, e.g. idioms and domain specific expressions, the actual number of word tokens in an utterance could often exceed seven.

There is a problem representing sequences of varying length in TiMBL, since the input format is of a fixed length. To get away from this problem, the semantic tags obtained from the speech recogniser lexicon, are represented as a binary feature vector, which means that the original tag sequences of the utterance hypotheses are lost. The assumption is that the sequencing of the semantic tags is not important as long as all tags are represented in the vector.

The analyser server

The analyser server consisted of a wrapper program around the TiMBL classifier (described above) and it ran under the same broker as all August components. The input to the server was an N-best list from the speech recogniser, and the output was a semantic analysis for each item in the N-best list. Each time the analyser server was started, it created the TiMBL training data from the example database and from the speech recogniser lexicon, thus keeping it up to date with any changes in the lexicon, which was a resource common to several components of the system. This was important from a robustness perspective, since the lexicon was being continually updated.

When the analyser server was started, it read four files: the three example utterance databases and a speech recogniser lexicon. Each item in the lexicon had a phonetic transcription, a grammatical category and a semantic tag. In addition to the lexical items, the analyser used the semantic tags.

Below, each sub-process of the semantic analyser is described.

Filtering out implausible hypotheses

The first slot of the semantic representation was a binary feature indicating whether a hypothesis from the speech recogniser seemed plausible or not. Speech recognisers often use a probabilistic language model, usually based on n-gram statistics. In the August project, a speech recogniser using a bigram grammar was used. This means that the recogniser could only "see" two words at a time, and was not always able to cope with certain long distance dependencies. Sometimes, rather strange hypotheses could find their way up to the top of the N-best list.

As a means of lessening this problem, one of the classifiers in the semantic analyser recognised implausible hypotheses–utterances that a human annotator had deemed impossible or highly unlikely. The annotator was given the result from running several hundred utterances through the speech recogniser, and all unique hypotheses in the resulting N-best lists were categorised as either acceptable or not. This approach was chosen as a simple and fast way of identifying strange hypotheses, without having to do anything about the language models the recogniser used. There was no clear-cut definition of what an impossible utterance was, but it was based on semantic grounds rather than grammatical ones only. For instance, number or gender agreement errors did not necessarily disqualify a hypothesis, while e.g. a semantically strange combination of verb and argument might do so. The point of judging the acceptability of an utterance hypothesis as a process separate from the rest of the analysis, is that of robustness: even though an analysis is deemed implausible, the rest of the semantic analysis might be of some use to the dialogue manager.

On an average, the classifier that identified implausible utterance hypotheses correctly classified an utterance for which there was no exact match in the training data 93% of the time. (In a "real world situation" there should hopefully be exact matches as well.) This figure was obtained by leaving out ten percent of the training data and testing it against the remaining training data. This was repeated ten times by randomly drawing the test data from 2,097 examples.

Topic prediction

The second slot of the semantic representation was instantiated with the topic of an utterance. There were several different domains or topics: Stockholm, Strindberg, yellow pages, speech technology, main and meta. The "main" domain included "social interaction", such as greetings and dialogue initiation, which did not belong to a specific domain. It also included handling assorted common questions, which did not belong to a specific domain. "Meta" included meta questions about the system itself and to some extent user reactions to system failures. The other domains allowed the user to ask questions about restaurants, the city of Stockholm, August Strindberg and about speech technology.

In an evaluation of the topic prediction carried out on a subset of the database of manually annotated user utterances, it turned out that about 71% of the utterance hypotheses were correctly classified (4,236 out of 5,926). It should be noted that many of the utterance hypotheses had been classified as implausible by the classifier described in the subsection above, and were for this reason potentially hard to classify as dealing with a certain topic.

Semantic feature-value pairs

The third field of the semantic representation consisted of feature-value pairs representing the utterance and was used by the dialogue manager to interpret an utterance in order to be able to do database look up, etc. In the "meta" domain, a question such as "how does this thing work?" might be given the feature value-pairs `{obj:august, function:?}`, while "how do I get to Hornsgatan?" or "show me the way to Hornsgatan" in the "yellow pages" domain would result in `{obj:map, place: hornsgatan}`.

Due to time constraints, it was not possible to manually annotate the user data with the semantic representation used by the analyser, thus making a proper evaluation hard. Furthermore, it is not obvious how such an evaluation should be done, since it was an unconstrained dialogue, where users often talked about out-of-domain topics. This fact, along with a hard acoustic environment, made the speech recogniser produce strange hypotheses, which the semantic analyser could not translate into the correct feature-value representation. However, by making use of the filter described above, the analyser could recognise many of these implausible hypotheses.

Related work

In Boye et al. (1999), a hybrid approach to semantic analysis in a spoken dialogue system in the travel-planning domain is presented. The output from the speech recogniser simultaneously underwent a deep semantic analysis, resulting in an interpretation in first-order logic, and a shallow slot-filler analysis. The slot-filler approach was added as a faster and more robust complement to the deep processing (which was also reduced to a flat representation). The hand-coded shallow semantic parser analysed the top utterance hypothesis from an N-best list by doing key word spotting. Other than producing lists of slot-filler pairs, the robust parser recognised utterance types, e.g. yes/no questions, wh-questions, etc.

Yet another system which followed two tracks of semantic analysis, one deep and one shallow, is reported in Kipp et al. (1999).

In Waxholm (Carlson et al., 1995), a system for travel information about boat trips in the Stockholm archipelago, the results from the speech recogniser were processed by a probabilistic syntactic parser which assigned to each hypothesis a syntactic analysis including semantic features. The parse tree was reduced to a semantic structure to instantiate slots of semantic templates. The parser also produced a topic prediction, based on semantic feature analysis.

In Olga (Beskow & McGlashan, 1997), a multi-modal consumer information service, a shallow semantic representation similar to the one of the current work was produced by running the output of a speech recogniser through a syntactic analyser and match the resulting dependency structure against a database of hand-coded examples. If no complete match was found, the analysis was built up from dependency tree fragments.

Discussion and future work

The analyser server was running for six months in a dialogue system available for the general public, and seemed to meet the requirements of robustness and to be easy to extend.

Initially, the analyser component described above was considered a mock-up while putting the system together, and it was felt that it should be replaced by a more advanced semantic interpreter as the system was being developed. However, it turned out that the mock-up suited our needs surprisingly well, and so the analyser running TiMBL as its core process was kept. One of the advantages of this approach was that, since only a fairly simple analysis was required, it was easy to extend the coverage of the analyser without any need for e.g. grammar or tree-bank development. No complex lexicon development was needed either, since the analyser used the very same lexicon as the speech recogniser did.

The issue of knowledge acquisition is very important if one wants to be able to swiftly incorporate new domains or extend existing ones. We believe that annotating actual examples produced by a speech recogniser with a flat semantic representation, without minding grammatical or lexical issues, is a fairly straightforward and quick way of developing a semantic component.

In some cases, the output of the analyser had to be post-processed. This was because some of the feature-value pairs could not be instantiated simply by classifying an utterance. One example of this was when the semantic representation of an utterance should return an instance of a category that must not necessarily be in the training data, such as street names. Perhaps only a few instances of utterances including street names are needed to correctly classify these utterances, thus excluding most of the possible street names from the training data. In these cases, the analyser returned e.g. a feature `STREET_NAME` and the postprocessor had to screen the input utterance for the particular instance of `STREET_NAME`. Future work includes automating this process. Furthermore, it might be worthwhile to consider *active learning*, as a means to speed up the annotation of the training database (see e.g. Thompson et al. (1999)).

References

Bell L & Gustafson J (1999a). Interaction with an animated agent in a spoken dialogue system. In: *Proceedings of Eurospeech 99*, 3: 1143-1146, Budapest, Hungary.

Bell L & Gustafson J (1999b). Utterance types in the August system. In: *Proceedings of the ESCA Workshop on Interactive Dialogue in Multi-Modal Systems (IDS'99)*, Kloster Irsee, Germany.

Beskow J (1997). Animation of talking agents. In: *Proceedings of AVSP'97, ESCA Workshop on Audio-Visual Speech Processing*, Rhodes, Greece.

Beskow J & McGlashan S (1997). Olga - a conversational agent with gestures. In: *Proceedings of the IJCAI'97 Workshop on Animated Interface Agents -- Making them Intelligent*, Nagoya, Japan.

Boye J, Wirén M, Rayner M, Lewin I, Carter D & Becket R (1999). Language-processing strategies for mixed-initiative dialogues. In: Alexandersson, J, editor, *IJCAI'99 Workshop Notes NLP-2: Knowledge and reasoning in practical dialogue systems*, Stockholm, Sweden; 17-24

Carlson R, Granström B & Hunnicutt S (1990). Multilingual text-to-speech development and applications. In: Ainsworth W, editor, *Advances in speech, hearing and language processing*, London: JAI Press; 269-296.

Carlson R, Hunnicutt S & Gustafson J (1995). Dialogue management in the Waxholm system. In:

Proceedings of Spoken Dialogue Systems, Vigsø, Denmark.

Daelemans W, Zavrel J, van der Sloot K & van den Bosch A (1998). *TiMBL: Tilburg Memory Based Learner, version 1.0, Reference Guide.* ILK Technical Report 98-03, Available at `http://ilk.kub.nl/~ilk/papers/ilk9803.ps.gz`.

Gustafson J, Lindberg N & Lundeberg M (1999). The August spoken dialogue system. In: *Proceedings of Eurospeech'99*, Budapest, Hungary.

Kipp M, Alexandersson J & Reithinger N (1999). Understanding spontaneous negotiation dialogues. In: Alexandersson J, editor, *IJCAI'99 Workshop Notes NLP-2: Knowledge and reasoning in practical dialogue systems*, Stockholm, Sweden; 57-63.

Ström N (1997). Automatic continuous speech recognition with rapid speaker adaptation for human/machine interaction. *PhD thesis,* Royal Institute of Technology, Stockholm, Sweden.

Thompson CA, Califf ME & Mooney RJ (1999). Active learning for natural language parsing and information extraction. In: *Proceeding of the Sixteenth International Conference on Machine Learning*, Bled, Slovenia.

Öhman T (2000). Vision in Speech Technology. *Licentiate thesis,* Royal Institute of Technology. Stockholm, Sweden.