Dialogue-based disambiguation

Koeling, Robert Wietse

Chapter 6

Conclusions

In the second chapter of this thesis I presented the state of the art, after about two years of work, of the natural language processing component we developed for the OVIS spoken dialogue system. During those two years a lot of effort was put in building a grammar which on the one hand can serve as a basis for a general computational grammar for Dutch and on the other hand allows for robust and efficient parsing algorithms for processing spoken utterances. We have demonstrated that these two requisites are not mutually exclusive. The evaluation of the NLP component shows that the accuracy results were at least as good as could be expected from 'simple and effective' approaches, without losing out on efficiency. Even though we could not prove that this more fundamental approach results in better accuracy scores (for this application), we think that it is an important step forward that a complete domain-independent structural analysis of an utterance can be given instead of just the (highly domain dependent) semantic slot values normally returned by one of those 'simple and effective' approaches like 'concept spotting'. We expect that issues like portability to other domains and the possibility of extending the coverage of the domain can be addressed easier and in a more fundamental way when there is a firm linguistic basis for analysing natural language expressions.

Another point worth considering is the vast amount of knowledge about language use that is available now and is neglected by some approaches to NLP. Of course, when abstract, but very efficient language models (such as for example language models based on probability distributions that only consider the previous two words important for predicting the next word (i.e. trigram models)) outperform complex models that offer good explanations of how sentences can look like, but need a lot of computational power and are very sensitive to irregularities in language use, it is, especially in systems that are used in the real world, very attractive to choose the former alternative. But when we choose the simple, abstract models, we do neglect information that might help us to build a better language model.

Even though the evaluation results I mentioned above were very gratifying, there was still room for improvement. The upper bound for the results was

defined by assuming an oracle that finds the path in a wordgraph that is as close to the actually uttered sentence as possible. The lower bound was defined by the evaluation results of the NLP component using the acoustic evidence given by the speech recognizer, the probabilities given by the trigram language model and the results of parsing. All of these three knowledge sources are plausible from a human perspective. It does not need any explanation that it makes sense to take acoustic evidence into account. Even though n-gram models only consider very local information (and thus can not be used to model all linguistic events), they can describe information that is difficult to catch otherwise. People do have intuitions about the likelihood of even a short string of words. The fact that people have different expectations about the continuation of 'Chairman Bill ...' and 'President Bill ...' can not be explained on the basis of structural differences, but only on the basis of the frequency of examples of these strings. Finally, the fact that people tend to produce grammatical sentences (even though spoken language is less error-free than written language) suggests that utterances that can be given a proper analysis are to be preferred. In my search for extra knowledge sources that can help to reduce the error rate, I looked at examples where the evaluated *nlp* component got it wrong, but at the same time the oracle suggested that a better analysis was possible. In quite a few cases it became clear that the hypothesis that was chosen from the wordgraph did not make any sense as a response to the question the system asked. This seemed to be a potential knowledge source. Especially in a system initiative spoken dialogue system, it is easy to describe the dialogue context. Moreover, interpretation in dialogue context is plausible from a human perspective: we tend to expect that people respond to questions asked, and we have expectations about what answers are reasonable.

Ginzburg's dialogue model provides a good framework for describing the dialogue context in which a user utterance must be analysed. One of the central notions of his dialogue model is the list of *questions under discussion*. This is a list of questions raised in the dialogue so far that still can (or need) be addressed. I found that OVIS dialogues can be described in a natural way in terms of Ginzburg's dialogue model. Ginzburg uses his dialogue model to give an analysis of short answers, which are very typical for the OVIS domain. I have adapted this theory in such a way that expectations about what can be said at a certain point are expressed as constraints. When the theory predicts that a certain (short) answer is not interpretable, the analysis receives a penalty. In my experiments I found that it was possible to rule out some hypotheses that were obviously not possible in that context. In general, however, this strategy proved to be too rigid. Although many different responses to a question are *possible*, there is often a strong preference for some over the others and I think that that is a better way of thinking about it. Although the approach proposed in Chapter 3 does not rule out the possibility of incorporating knowledge that allows us to express a preference for utterances in certain dialogue contexts, I have decided not to continue in this direction. This is because I think that a theory like the one adopted later is more suitably regarded as a means of expressing expectations about what the next dialogue move can be, than as

a *filter* to be applied (filtering out those speech hypotheses that are not interpretable as I have implemented here). Expectations about dialogue moves could be used in a similar way as the parsing results to compare competing hypotheses. Statistical data about sequences of speech acts in these dialogues could then be used to quantify the preference of the possibilities.

I think the main drawback of my first approach is the lack of flexibility. In that approach an hypothesis is either acceptable or not, and there is no way of expressing a preference for a certain solution. Some way of integrating probability theory can help us to solve that problem. The question now is how to do this. Even though I think that a proper dialogue model can supply usable information for disambiguating speech input, I have chosen to integrate statistical information in a different way. This was mainly motivated by the second drawback of the previous approach: the difficulties introduced by adding another knowledge source. Adding another knowledge source in formula 5.1 can be avoided by replacing one of the components by the same component that is made sensitive to dialogue context. The most obvious candidate for this is the only probabilistic component available: the n-gram model.

It was mentioned before that n-gram models are amazingly effective, even for small n. For the experiments in the previous chapters I have only used trigrams. That means that only the two previous words are taken into consideration to estimate the probability of the next word. There is no inherent need to limit yourself to just the two previous words, but it is often the case that extending the window one or two words to the left does not help much, and it generally increases the model size enormously. In the last two chapters I investigated how the language model can be made sensitive to the dialogue context. The easiest way of doing this is to create a special n-gram model for every context. For the OVIS application this is also very much feasible. I have identified only a limited number of different contexts (namely the five types of system questions) and there is a lot of training material available. Training different trigram models for all five contexts brings the perplexity of the language models down, and from an engineering point of view this would be a very reasonable thing to do to improve the accuracy of this particular system. It is very unsatisfactory, though, to fragment the available training data, and a method that allows us to use all the available data irrespective of the number of contexts defined, would be a step forward to a general solution. A move in that direction can be made by defining a trigram model and a model that describes the relation between words in the utterance and information about the type of the preceding question. Interpolation of the two models gives us the advantage of making the resulting language model sensitive to dialogue context without fragmenting the training data. However, the integration of the two different knowledge sources is not optimal in the cases where the knowledge sources are not statistically independent. This is clearly the case for the two models we want to use here. The Maximum Entropy modeling framework addresses this problem successfully. It can cope with knowledge sources that partly describe the same information. The weights assigned to overlapping features will reflect this and the modeler

does not have to worry about this.

The MaxEnt context model I have developed in the last chapter integrates just the two knowledge sources mentioned above. Again, there is no need to limit ourselves to these two. One of the virtues of the MaxEnt framework is that it allows for combining all kinds of different knowledge. A third important virtue of MaxEnt models that needs to be mentioned here is the fact that MaxEnt models perform very well. The implementation of the idea that conclusions ought to be drawn on basis of evidence, leads to a formalism that automatically takes into account all the information available (i.e. selected information from a sample space), but avoids assuming information we do not have. Probability mass that is not accounted for by the data, is equally spread over the model (maximizing the entropy of the model). The improvement in performance due to the MaxEnt context model developed in the last chapter is gratifying. Although the absolute gain in error reduction is relatively modest, we have to be aware of the fact that the space for improvement is very limited (the gap between the lower and the upper bound was only 6%). So far I have only included one type of contextual information (the relation between type of system question and words in the user utterance). Even though I think it is the most suitable (at least for first experiments), because it is a very simple, well defined and informative knowledge source, extended views on dialogue context could help to improve the model. A second point I would like to make here is the fact that almost all of the changes triggered by the context model are improvements. This suggests that the model does not suffer from too much bias towards an hypothesis that fits well into the dialogue context.

Another positive aspect of the MaxEnt context model is that it is not designed only for this application. I have mentioned before that the OVIS application provides a particularly good environment for these experiments (system initiated dialogue, simple well structured dialogues and few different types of system questions). This does not mean, however, that the application of these models is restricted to systems with the above-mentioned characteristics. MaXEnt is capable of dealing with increased complexity of the dialogue, and a larger number of different dialogue contexts. Mixed initiative (instead of system-initiated) systems will be a bit more challenging because the dialogue state would no longer be given, but would have to be detected. The consequence of the added uncertainty of the dialogue state means that it will be harder to gain from the extra information source.

The virtues of MaxEnt models listed in section 5.2.6 open a range of possibilities for future research. The fact that many different, possibly unrelated knowledge sources can be integrated in one model suggests nice opportunities for experiments. In Chapter 5 I suggested, for example, to include more 'linguistically motivated' features. This could be done on the level of constituents that are meaningful for that particular application (such as *locative noun phrases* or prepositional phrases headed by a particular preposition might be ideas for the OVIS application). Added knowledge can also come from completely different

sources, however. As discussed before, a good dialogue model can supply useful information about next dialogue moves. That information might also be included. A completely different idea might be, for example, that when a system is intensively used by a small number of users, to *personalise* the language model by adding features that relate words to a particular speaker.
When a software package to calculate the MaxEnt models is available, experimenting is made easy by the nature of the framework. The modeler does not have to worry about conflicting knowledge sources. This means that a linguist without knowledge about statistical modeling can work on these models.

Another line of future research would concern MaxEnt models in general. Even though the framework has many good characteristics, it is still producing an *approximation* of the model that produced the sample data. The best way to get a closer approximation of the process we are trying top model, is to include more information about the process (i.e. add more (meaningful) features). Even though MaxEnt models are less prone to suffer from sparse data problems than many other frameworks, it is in the nature of corpus-based models to have problems with 'not enough examples' of certain events. In my experiments I have not worried about applying smoothing techniques to tackle this problem. There exists research about smoothing algorithms for MaxEnt models. It would be good to see if the models can benefit from those efforts. Finally, something has to be said about the major drawback of the framework: models tend to get big and (especially if we have to compute probabilities at run-time) can become slow in use. So far, not enough time has been spent on trying to minimise the models. Feature selection algorithms are an interesting area to investigate.