Iterative Statistical Language Model Generation for Use with an Agent-Oriented Natural Language Interface

Babak Hodjat

Dejima Inc.
160 W Santa Clara St. #102,
San Jose, CA 95113, USA
Babak@dejima.com

Horacio Franco
Harry Bratt
Kristin Precoda
Andreas Stolcke
Anand Venkataraman
Dimitra Vergyri
Jing Zheng

SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025-3493, USA
hef@speech.sri.com

Abstract

We describe a method for developing a statistical language model (SLM) with high keyword spotting accuracy for a natural language interface (NLI). The NLI is based on the Adaptive Agent Oriented Software Architecture (AAOSA). Our experience shows that this method provides for rapid development of an SLM that is well suited to the requirements of the agent-oriented NLI. Experiment results show a comparatively low equal error rate of 13.2% for a vocabulary of 2400 keywords. This result is a robust free-form speech-based NLI with a high task completion rate.

1 Introduction

In a natural language interfaces (NLI), natural language is the main mode of interaction with the back-end system. The fundamental principle in an NLI is to give the user freedom to express intentions without limitations. The NLI typically responds to the user in one of the following ways:

If the request is understood, is in the scope of the application, and is not ambiguous, it is mapped to one or more actions in the back-end system. If the request is understood and in scope but is ambiguous, the NLI either defaults to actions handled by the back-end, or interacts with the user to resolve the ambiguity. Discourse history or a statistical model of user preferences can drive the choice. If the request is understood as outside the scope of the back-end system, the NLI provides an explanation of the covered domain. If the request is not understood, the NLI either provides help, or gracefully downgrades to a keyword search on the data in the back-end system. For a survey of work on NLI's see Perrault and Grosz (1988), Rich (1987), Copestake and Sparck-Jones (1990), and Bates (1987).

NLIs often use text as their main input modality; speech is however, a natural and, in many cases, preferred modality for NLIs. Various speech recognition techniques can be used to provide a speech front end to an NLI. **Grammar-based recognizers** are rigid and unforgiving, and thus can overshadow the robustness and usability of a good NLI. **Word-spotting recognizers** are reliable only when the input consists of short utterances, and the number of words to be spotted at each given time is small. **Dictation engines** are processor and memory intensive, and often speaker

dependent. The dictation vocabulary is often considerably larger than required for domain-specific tasks. **General statistical language models (SLMs)**, although robust enough to be used as a front end for a structured domain, requires a very large training corpus. This is time consuming and expensive since a large number of users needs to be sampled and all speech has to be transcribed.

In this paper we propose a method for rapidly creating a suitable statistical language model for a recognizer that can be used as the speech input modality of an AAOSA-based NLI. We also provide some experimental evidence on its performance suitability.

2 AAOSA-NLI and Application

We use Dejima™ Direct® Platform, an NLI based on the Adaptive Agent-Oriented Software Architecture (see Figure 1) (Hodjat and Amamiya 2000a, 2000b). In this NLI, agents are defined as independent software entities that interact with each other by exchanging messages in an agent communication language.

Networks of agents are organized to address problems in a particular domain. The agents work as a group, cooperating, and competing, to produce a solution. The approach is non-centralized; sub-groups of agents are specialized in different aspects of the domain, work in parallel towards a solution. Ambiguities are resolved by the agents or through dialogues with the user.

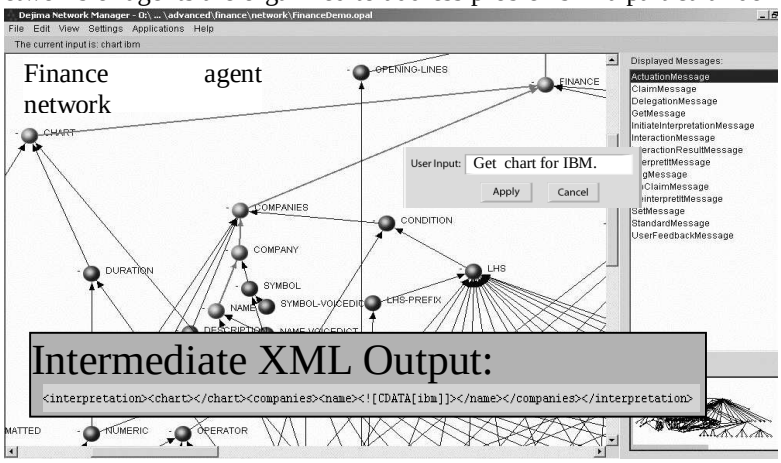


Figure 1: Agent network for a financial application. Arrows indicate the claims being propagated up-chain. Green agents are agents with winning claims. The XML is constructed from contributions generated by the winning agents.

An AAOSA-based NLI has certain advantages, including:

- Principled use of context for disambiguation: Ambiguity resolution built into AAOSA.
- Learning capabilities: This feature is not easy to use with a speech modality since speech recognizers have constraints on learning previously un-encountered words.
- Natural interaction for partially completed requests: Users get prompted for clarification to resolve ambiguities or to complete a request.
- Concurrent integration methodology: Concurrent actuation of partial requests, no need to complete the command for the application sequentially (i.e., the pipeline approach).
- Extensibility: This follows from the integration methodology noted above. Adding new agents has a minimal disruptive effect on the behavior of existing agents.
- Knowledge engineering equals language engineering: The agent network is a semantic description of the application domain language from the user's perspective;

A finance application with an AAOSA NLI was used for the evaluation (http://www.dejima.com/demos.html). The application answers requests about information on S&P 500 companies, such as current price, annual high and low prices, dividend, volume, and yield, (e.g., "What is IBM trading at?"). The information is presented as text, tables or charts, depending on a user's request (e.g., "Volume for Sun", "Volume and dividend for Sun and Coke, and "Plot SUNW vs. INTC"). Context can be used to add follow-ups such as more companies and parameters to a given query (e.g. "Compare that to AMD")

3 Speech recognition method

Our approach is to do wordspotting of the vocabulary that is relevant to the finance application with the AAOSA NLI, and to ignore words outside that vocabulary. This approach enables the user to speak in a natural fashion, without being constrained by menus or system prompts.

The number of relevant words, or "keywords", was in the order of a few thousand. This made it impractical to use traditional wordspotting approaches. For our approach, we developed a variation of the following wordspotting method developed at SRI. The SRI wordspotter is based on a large-vocabulary continuous speech recognition (LVCSR) engine (Weintraub, 1995). The LVCSR system produces an N-best list of likely recognition hypotheses, which are then post-processed to obtain posterior probabilities (confidence levels) for individual words. Based on the output, an individual word is detected (spotted) when its probability is over a predefined threshold. We made the following changes to the SRI wordspotter, First, the N-best lists were post-processed into word confusion networks (Mangu et al. 2000), which are known to give more accurate word recognition and posterior probability estimation (Stolcke et al. 2000). Second, we used a statistical language model that requires significantly less training data than an LVCSR system. A key aspect of the problem is that the number of keywords, while high for a traditional wordspotting system, is still smaller than the vocabulary of an LVCSR or dictation system. To target this type of application, our approach was based on the following three ideas: (i) combine the predictive power of statistical language models with the use of filler models to reduce the size of the active vocabulary while keeping the grammar flexible; (ii) use word classes to reduce the number of different tokens for which the statistical language model needs to estimate probabilities; and (iii) develop domain-specific SLMs to reduce the vocabulary size as well as the grammar perplexity. This combination of features allowed us to build SLMs with significantly fewer training sentences than a more standard SLM would require.

We implemented a medium-vocabulary, speaker-independent continuous recognition system with a vocabulary formed as follows: first, we included all the relevant keywords, secondly, we added a set of frequent words that are not keywords but are likely to appear in naturally constructed sentences; and third, we used a filler model to represent all other out-of-vocabulary words. An interesting feature of this approach is that in addition to the filler model, acoustic models are created for frequent non-keyword words. These non-keywords are integrated into the grammar constraints embodied by the SLM. The non-keyword models jointly function as a detailed acoustic filler model. Another interesting feature resulting from this approach is that successive words mapped to a filler model are merged into a single filler token. Thus, a statistical language model based on trigrams, or higher-order statistics, can capture context across out-of-vocabulary regions, effectively allowing long-distance dependencies in the language model.

The class-based N-gram language model was trained using the SRI Language Modeling Toolkit (SRILM; Stolcke 2002). Word classes were domain-specific and carefully chosen by hand. We defined separate word classes for stock names, company names, and index names, as well as for

cardinal, ordinal, and time expressions. For SLM training, the word tokens belonging to the classes were replaced by the corresponding class labels. During recognition, classes were identified and expanded dynamically in the search. Class membership probabilities were also estimated from domain-specific data. For example, the probability of a specific company name was estimated from the volume of its shares traded over the previous month.

As an enhancement step, generic SLMs derived from large general corpora (such as the LDC Broadcast News corpus) were combined with the domain-specific SLM. However, we did not observe improvements with this approach and abandoned it for now. We believe that this negative result is due to the significantly different nature of the test data and the generic models. To improve the quality of the acoustic modeling, common company names were converted into multi-words to enable efficient use of more accurate crossword triphone models; this approach also simplifies the detection procedure for multi-word company names.

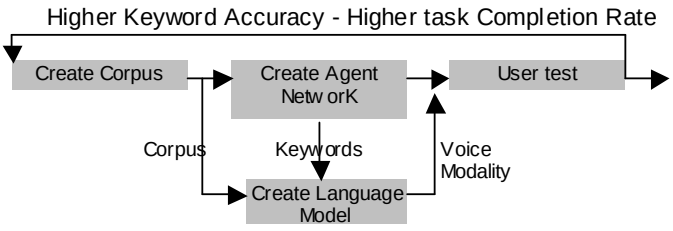
4 Experiments

A set of 320 spoken commands and sentences was collected from ten users (five male, five female) in a Wizard of Oz setting (Dahlback et al. 1993) and was used as the test set in the experiments described below. Recognition output in the form of N-best hypotheses, with N=1000, was obtained, from which we generated word confusion lattices with posterior probabilities for individual words using the SRILM toolkit.

The system ran at close to real-time on the SRI Dynaspeak engine (Franco et al. 2002) on a 1-GHz Pentium-III personal computer using a class-based trigram language model. We generated the receiver operating characteristic (ROC) curve for keyword detection. The ROC curve was used to evaluate wordspotting performance and to define the desired operating point, that is, the balance between the probability of mistakenly spotting a keyword that did not occur, versus the probability of missing a keyword that did occur. The corresponding equal error rate (EER), when both types of errors are equated, was 13.2 % for a vocabulary size of 2400 keywords and 600 additional non-keyword words. The integrated speech-based financial NLI was tested using five male and five female test subjects. Each subject formulated 75 commands in response to given scenarios, and also uttered an additional 25 free-form commands. A task completion rate of 92% was achieved.

5 Iterative System Development

Figure 2 shows the development steps of an SLM for an AAOSA-NLI. A text corpus of sample user queries and interactions are collected to develop an agent network.



This corpus is also used to train the SLM. After a first pass at creating the agent networks, words and phrases that play an important role in agent interpretations are extracted as a keyword list to be included in the wordspotting language model.

Figure 2: Creating a statistical language model for the AAOSA-NLI fits well into the development steps of the NLI itself.

This helps the speech recognizer to improve recognition of in domain keywords. The resulting recognition system provides the agent network with a speech input modality used in the second

pass of data collection. This pass helps refine both the agent network and the statistical language models, and is also used for collecting spoken sample utterances to further develop and tune the system.

6 Conclusion

Pairing a statistical language model and an AAOSA NLI provides for a robust interface in which users can use utterances that are not grammatically complete (e.g., "IBM quote") or that comprise multiple commands (e.g., "Quote IBM, AT&T, and HP, and chart them and get all companies with pe > open"). The AAOSA-NLI recovers 50 to 70% of utterances containing misrecognitions and drastically increases task completion rates due to dynamic management of ambiguities within the AAOSA-NLI.

References

Bates, M. (1987), Natural-Language Interfaces. In S.C. Shapiro & D. Eckroth (eds), *Encyclopaedia of Artificial Intelligence*, Wiley, New York.

Copestake, A. and Sparck-Jones, K. (1990), Natural Language Interfaces to Databases. Technical Report 187, Computer Laboratory, University of Cambridge.

Dahlback, N., Jonsson, A., and Ahrenberg, L. (1993), Wizard of Oz Studies – Why and How, *Intelligent User Interfaces '93*, ACM.

Franco, H., Zheng, J., Butzberger, J., Cesari, F., Frandsen, M., Arnold, J., & Gadde, V. R. R., Stolcke, A., and Abrash, V. (2002), DynaSpeak: SRI's Scalable Speech Recognizer for Embedded and Mobile Systems. *Proc. Human Language Technology Conference*, San Diego, CA.

Hodjat, B. and Amamiya, M. (2000a), Applying the Adaptive Agent Oriented Software Architecture to the Parsing of Context Sensitive Grammars, *IEICE Transactions on Information and Systems*, vol. E83-D, no. 5, pp. 1142-1152, Japan.

Hodjat, B. and Amamiya M. (2000b), Introducing the Adaptive Agent Oriented Software Architecture and Its Application in Natural Language User Interfaces, in *Agent-Oriented Software Engineering*, Springer.

Mangu, L., Brill, E., and Stolcke, A. (2000), Finding consensus in speech recognition: word error minimization and other applications of confusion networks, *Computer Speech and Language* 14(4), 373-400.

Perrault, C.R. and Grosz, B. J. (1988), Natural Language Interfaces. In H. E. Shrobe, *Exploring Artificial Intelligence, Survey Talks from the National Conferences on Artificial Intelligence*, Morgan Kaufman, CA.

Rich, E. (1987), Natural-Language Interfaces. In R. M. Baecker & W. A. S. Buxton (eds.), *Readings in Human-Computer Interaction*, Morgan Kaufmann, San Mateo, CA.

Stolcke, A. (2002), SRILM – An Extensible Language Modeling Toolkit, *Proc. Intl. Conf. on Spoken Language Processing*, vol. 2, pp. 901-904, Denver, CO.

Stolcke, A., Bratt, H., Butzberger, J., Franco, H., Gadde, V. R. R., Plauche, M., Richey, C., Shriberg, E., Sonmez, K., Weng, F., and Zheng, J. (2000), The SRI March 2000 Hub-5 Conversational Speech Transcription System, *Proc. NIST Speech Transcription Workshop*, College Park, MD.

Weintraub, M (1995), LVCSR Log-likelihood Ratio Scoring for Keyword Spotting, *Proc. IEEE Intl. Conf. on Speech and Signal Processing*, vol. 1, pp. 297–300, Detroit.