An Adaptive Speech Recognizer that Learns using Short and Long Term Memory

Helen Wright Hastie and Jody J. Daniels

Lockheed Martin Advanced Technology Laboratories
3 Executive Campus, 6th Floor
Cherry Hill, NJ 08002
{hhastie, jdaniels}@atl.lmco.com

Abstract

The world is constantly changing and new words and concepts are continually being created. Therefore, Spoken Language Systems need to be able to function in a dynamic environment and adapt. In this paper, we propose a number of ways in which the supervisor can modify the recognizer without the need for costly, esoteric developer knowledge.

1 Introduction

As the world changes and new words and concepts are created, Spoken Language System (SLS) can no longer be static. They need to be able to adapt to a changing environment and the user needs to be able to initiate this adaptation easily, without the need for costly developer intervention. In this paper, we concentrate on developing a recognizer that has flexible and dynamic vocabulary. In current technology, the recognizer has a fixed list of vocabulary. The order of words is predicted using either an inflexible grammar or a stochastic language model trained on typical data for that domain. Once the recognizer has been trained and deployed it cannot be changed in order to include new words, such as a new restaurant name, thus risking a degradation of accuracy over time. In this paper, we address the issue of how the supervisor can change and update the recognizer during a conversation.

There are a number of ways in which the supervisor can intervene to include new words or concepts into the recognizer. In order to obtain the spelling and pronunciation of a word, the user can say and spell the name; the user can spell out the word using the phonetic alphabet (alpha, bravo etc.); thirdly and most difficultly, the system can automatically recognizer that the word is missing from its vocabulary. There are a number of drawbacks with the first two of these approaches. Firstly, the user may have difficulty pronouncing or spelling the word. Secondly, this process is time consuming and may be annoying to the supervisor. In our experiments, we deploy the second tactic of having the user spell the missing words phonetically which the military are used to doing, for example with names. However, we will also describe techniques that work towards the goal of fully automating the process such as automatically obtaining the pronunciation through letter to sound rules.

We propose two approaches to automatically adapting the recognizer. The first takes into account information given by the supervisor within the same conversation. This information can be used later on in the dialogue to improve performance of the recognizer. Secondly, the more the supervisor uses the system the better the performance will be. This is facilitated by the long term memory which is used to learn from previous mistakes gathered across a number of dialogues.

These approaches are seemlessly integrated into the conversation, thus allowing supervisors to change the nature of the recognizer and improve its performance in an easy and natural manner.

1.1 The Challenge

Unlike humans, current speech recognition techniques are not able to utilize experience gained through previous interactions or from one part of the dialogue to the next. The example below illustrates how general knowledge and short term memory are used.

- John Doe was injured on September 18th, 2003 in Afghanistan.
- The Sergeant was driving a HMMV in Kabul when he had an accident.

The listener uses short term memory to infer that "John Doe" and "The Sergeant" are co-referents. He also uses long term memory that Kabul is in Afghanistan. An ideal system should be able to encode this type of knowledge in order to fill in gaps and reduce the search space by restricting it to relevant vocabulary and concepts.

2 The Corpus

The chosen domain for this work is military casualty reporting. Our recognizer is part of a Spoken Language System (SLS) system to assist military personnel in reporting battlefield casualties directly into a main database. The dialogue consists of two parts. The first part is a data gathering process, whereby the speaker gives information, such as casualty name, social security number, unit affiliation etc. The second part of the dialogue is more open-ended and free-form when the speaker describes the circumstances of the

casualty. The work discussed here focuses on this last part of the dialogue, which contains very important information but which is also the most difficult to recognize.

Although the vocabulary of the test data is of a reasonable size (1000 words), there is little overlap between the training and the test data. 34% of the vocabulary in the test set is not in the training set, we refer to these words as "out of vocabulary" (OOV). This is reflected in a high trigram perplexity of 10.6. Perplexity is used frequently in automatic speech recognition to signify the difficulty of the task. It can be thought of as the weighted average number of choices a random variable has to make. High perplexity indicates a low predictability and indicates that this is a difficult recognition task. Indeed our baseline recognizer has a word accuracy of 53% which is trained on the 75% percent of the data and tested on the remaining 25%.

3 System Architecture

The Casualty Reporting Spoken Language system uses the Galaxy architecture (Seneff, Lau, & Polifroni 1999). This Galaxy architecture consists of a central hub and servers. Each of the servers performs a specific function, such as speech recognition, understanding, context tracking, turn manager, text-to-speech, etc. The individual servers exchange information by sending messages through the hub. These messages contain information to be sent to other servers as well as information used to determine what server or servers should be contacted next. This spoken language system is used to gather the initial information about the casualty by filling out a virtual form. The system is mixed initiative, for example the user is able to fill out a number of fields in one utterance at any time in the dialogue. The system will prompt the user for any missing information. It explicitly confirms any information that the user provides and the user can correct any mistakes, rendering this information gathering process highly accurate.

The task of making an entire SLS system learn and adapt is a large area of research. This paper looks specifically at augmenting the recognizer server. The baseline circumstance recognizer used is SUMMIT, developed at MIT (Strom *et al.* 1999; Glass, Chang, & McCandless 1996). This speech recognizer uses a segment-based approach for modeling acoustic-phonetic events and utilizes Finite-State Transducer (FST) technology to efficiently represent all aspects of the speech hierarchy including the phonological rules, the lexicon, and the probabilistic language model. This language model is a combination bigram-trigram, class-based model. The output of the recognizer is a list of n-best hypotheses of what the speaker said.

Figure 1 shows how we take this baseline recognizer and add two types of resources: short and long term memory. The short term memory consists of the casualty data extracted from the first part of the dialogue. This process involves a boosting technique that takes salient lexical items from the casualty data and boosts the likelihood in the recognizer. Boosting certain words makes it more likely that the recognizer will pick this word over a similar sounding, less relevant word. One can think of this as changing the likelihoods in the recognizer to adapt to each circumstance data.

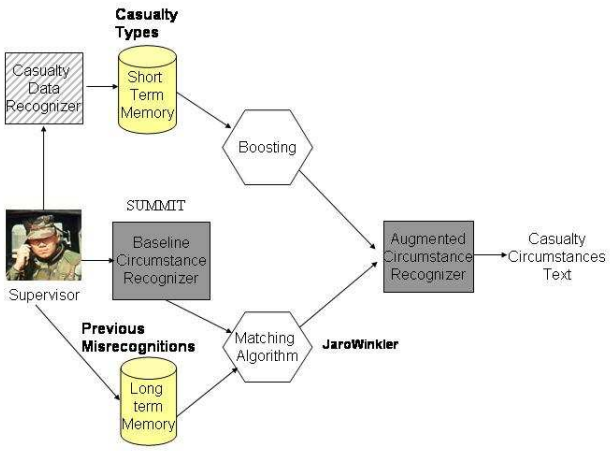

Figure 1: Using long and short term memory

The second source of information is the long term database. We take the most likely hypothesis from the baseline circumstance recognizer and use a matching algorithm to replace misrecognitions with more likely candidates. Each of these processes are described in detail in the following two sections.

4 Using Short Term Memory

In the first part of the dialogue, the supervisor provides key information such as name, rank, SSN, location of casualty, etc. Our hypothesis is that words and concepts used in the first, form-filling part will be repeated in the circumstances part. As mentioned above, we deploy a technique called boosting that involves increasing the likelihoods of words identified in the first part of the dialogue, during recognition of the circumstances.

4.1 Boosting Technique

The form-filling dialogue was recognized using a smaller vocabulary recognizer, trained specifically for this part of the dialogue. This recognition process is highly accurate, due to explicit confirmations and user corrections that are incorporated into the dialogue design. Out of vocabulary words, such as names, are obtained by spelling using military phonetics, e.g. Jones is said as "juliet oscar november echo sierra". In order to include these new words in the circumstance recognizer, we have to add a phonemic representation and make sure it is properly represented in the language model. This type of phonetic spelling is very familiar to military users in the field of casualty reporting, no pronunciation of the word is needed. An alternative is to let the user say the word naturally and use an out-of-vocabulary (OOV) recognizer (Gallwitz, Noth, & Niemann 1996) that would identify that this word is missing from the vocabulary. However, out-of-vocabulary detection is beyond the scope of this paper.

Text-to-speech (TTS) letter to sound rules from the Festival Speech Synthesis system (Black, Taylor, & Caley 1999) are used to automatically generate the baseform or phonemic transcription. This automatic technique was compared to one using hand-transcribed phonetic baseforms and both yield approximately the same results.

The new words are added to the circumstance training data multiple times so that the recognizer associates reasonably high likelihoods with them. Boosting is this process of adding an item multiple times into the training data and involves some trial and error in order to achieve optimal coverage in the language model.

4.2 Results

If you recall, the hypothesis behind using the short term memory populated by the supervisor, is that the data collected in the first part of the dialogue will reappear in the circumstance description. In order to test this, we investigated one data type, namely rank and names.

Table 1 gives the bigram and trigram perplexities and the corresponding overall word accuracy of the different types of recognizer. The first line is the baseline recognizer that is trained on the original training set. The high perplexity and low accuracy of the baseline recognizer, given on the first row of the table, indicates that the training set does not tell us much about the test set. The second recognizer is the first with the addition of the automatically generated baseforms of the new name. The third line gives results for retraining the recognizer, in order to boost the name associated with each circumstance.

Here, we can see that simply adding the names into the vocabulary improves this coverage and the overall word accuracy from 53% to 55%. Finally, we looked at boosting the likelihoods of the relevant name in the recognizer which results in a word accuracy of 58.8% which is a significant improvement over the baseline, by a paired t-test. These results are using the hand-transcribed baseforms. However, using the automatic baseforms does not reduce the accuracy significantly, from 58.8% to 58.7%.

	Bigram	**Trigram**	**baseline**
	Perplexity	**Perplexity**	**accuracy**
Baseline recognizer	528	18.3	52.9%
Name added	74.2	3.6	55.1%
Name boosted	31.5	2.6	58.8%

Table 1: Name recognition

Table 2 gives the recognition accuracy of just the names and name and rank using hand-written phonemic transcription. Name recognition is increased from 49% to 82% by simply adding the name into the vocabulary list. This figure is increased to 96% when this lexical item is boosted. The results for name and rank recognition are generally higher because the rank lexical items are already reasonably well represented in the training data. Boosting name and rank results in a further increase to an accuracy of 98.4%. Using automatically generated phonemic transcription only reduces the result from 82% to 79.4% for the "all names added" recognizer.

	Name	**Name and Rank**
no names	49%	76%
all names added	82%	91%
specific name boosted	96%	98.4%

Table 2: Name and recognition accuracy

These results indicate that our hypothesis is true and that salient words in the first part of the dialogue, do indeed reappear later on and that our boosting technique can result in a significant improvement of recognition accuracy.

5 Using Long Term Memory

A Spoken Language System should be able to learn and improve the more it is used by the supervisor. In order to emulate long term memory, we created a database that contained a set of previously misrecognized strings. We used machine learning techniques and distance metrics to perform matching on the hypothesized string to entries in the database.

If we think that a string is being misrecognized then the system will do a second run and substitute the hypothesis for a more likely candidate. Table 3 gives possible confusion pairs such as "the soldier fried shots" which is an unlikely utterance given the context and is substituted for "the soldier fired shots". Candidates for this memory based replacement could be determined by a number of algorithms from simple ones based on the recognizer likelihood (Gokhan Tur 2003) to more complex ones based on studies that identify misrecognitions based on prosody (Litman 1999). The development of these algorithms is outside the scope of the current study.

Recognizer Output	**Correct utterance**
the soldier shot head	the soldier was shot in the head
the soldier fried shots	the soldier fired shots

Table 3: Example long term database entries

5.1 The Match Databases

Ideally, this database would be developed during the deployment of the SLS, whereby misrecognition candidates would be hand-corrected and entered into the database. Unfortunately, our casualty reporting system has not yet been deployed, therefore, we had to simulate this database. One can also think of this database as a method of correcting any common misrecognition that the supervisor notices constantly occuring. This would simply involve entering the common mismatch into the database.

For our experiments, in order to fill the database with entries that sound similar and could be easily confused, we

used other hypotheses produced by the recognizer. Recognition candidates that occur further down in the n-best list are easily confusable as they are the next best match.

Two variations of this database were created: one at an utterance level and one at an utterance fragment level. The utterance level contains whole hypotheses from the recognizer in either words or phones. Entries in the utterance fragment database may consist of single words or groups of words that are consistently misrecognized. This database was formed by first chunking the utterance database into separate syntactic chunks using LT_CHUNKER from the University of Edinburgh (David McKelvie & Thompson 1997) and identifying chunks that contain misrecognitions. The matching algorithms for these two types of databases are described below.

5.2 Utterance Level Matching

In order to find the nearest match between our top hypothesis and an utterance in the database a number of machine learning techniques were deployed. The first of these is a rule-based building technique called RIPPER (Cohen *et al.* 1997). This technique was chosen over similar techniques, such as Classification and Regression Trees (CART) as RIPPER is able to handle a 'bag of words', where other techniques require that you specify each variant. This technique correctly mapped the top hypothesis with corresponding ones in the database 90% of the time, increasing word accuracy from 53% to 89%. An example rule from RIPPER is given below:

- **Recognizer:**..afghan town of gardez southeast of kabul..
- **Ripper:** if utterance contains "campbell" and "gardez". This indicates that "kabul" is misrecognized as "campbell"

Other string matching techniques were employed including calculating the JaroWinkler distance between the two strings of phones. Calculations were performed using Cohen's Secondstring program (Cohen, Ravikumar, & Fienberg 2003). This method yields an increase in word accuracy from 53% to 82%. RIPPER yields word accuracy results that are slightly higher, in addition this method is computationally less expensive.

One drawback with using this database is that it assumes that we are likely to encounter whole utterances in exactly the same form more than once. In fact, it is more likely that we will encounter parts of utterances. This is addressed in the following section.

5.3 Utterance Fragment Matching

The table below gives an example utterance fragment database. This includes the misrecognized words as well as its previous context in brackets. For each misrecognized word or group of words, we performed an exact match on the database entry. We then replaced the misrecognized word with the one in the same context. Using context allows us to disambiguate between words with double entries, such as the word "Campbell" which should be "Kabul" in one context and "camp"in another.

Recognizer Output	Correct utterance
[We drove to] Campbell	[we drove to] Kabul
[We made] Campbell	[we made] camp

Table 4: Example of fragmented long term database entries

This method of substituting misrecognition candidates by entries in the database, yields a significant improvement in word accuracy from 53%-67% (by a paired t-test). As the system develops over time, the growth of this database will taper off as the system reaches optimal performance.

6 Related Work

There have been a number of studies that look at all aspects of learning and supervisory learning in the different parts of a spoken language system. These studies vary from learning to adapt to different background noise using signal processing techniques (Yen & Zhao 1996) to reinforcement learning for dialogue strategies (Walker 2000). In this section, we concentrate on the use of memory and learning techniques for automatic speech recognition and information retrieval algorithms.

(Chung & Wang 2003) have developed a program that allows the user to add new vocabulary items online by saying and spelling a word using a program called ANGIE. With this software, the user has to speak and spell the name using the alphabet. This lexical item and phonetic transcription are automatically inferred and added into their recognizer dynamically. The advantage of this software is that it provides an automatic baseform transcription of the new word which can then be added into the recognizer baseform.

Our work using long term memory was inspired by machine translation techniques (Arnold *et al.* 1994) which use a database of frequently seen mappings from one language to another. Here we use a similar look up technique that searches for matches in a database of previous misrecognitions.

(Logan & Thong 2002) use a look up database similar to our long term memory for information retrieval query expansions. They use the recognizer to develop a table of words and phrases that are likely to be misrecognized due to their phonetic similarity. They use this database of confusable phrases to query their retrieval index, searching for exact matches of each phrase.

Constant use of a live system requires that the system be continuously evaluated and updated, to make sure that it does not become archaic and continually improves. (Gokhan Tur 2003) use active learning techniques in their system of automatic utterance or call classification to determine which utterances are the most interesting. They use two methods, the first is inspired by certainty-based active learning and selects the examples that the classifier is least confident about. The second method is inspired by committee-based active learning and selects utterances that the different classifiers do not agree on. Their interesting utterances are then examined by hand and used to update

the call classifier. This method reduces the human labelling effort by at least a factor of two.

7 Discussion and Future Work

This paper has presented a method by which the supervisor can improve the accuracy of a speech recognizer in an easy and seemless manner. The speech recognizer is a single part of the Spoken Language System, many parts of which could also benefit from becoming dynamic, adaptable and easily transferable from one domain to the next. For example, an area of future research would be to develop a stochastic turn manager that can train itself given human-human conversation or dynamically adapt given changing style of human-computer interaction.

Other areas where the system could adapt is in its relation to the supervisor. The system could upload supervisor profiles containing information from acoustic models, to user preferences and security clearances. The system could also infer information such as priorities, user specific phrases, expertise and frequency of use, speaker identification and verification, etc.

8 Acknowledgments

We would like to acknowledge the US Army at Fort Jackson for providing casualty reporting data. As an affiliate, we would also like to acknowledge MIT for providing us with their Galaxy II Spoken Language System. We would like to thank Steve Knott for his work on the experimentation.

References

Arnold, D. L.; Balkan, R.; Humphreys, S. L.; Meijer, S.; and Sadler, L. 1994. *Machine Translation. An Introductory Guide*. Manchester-Oxford: NCC Blackwell.

Black, A.; Taylor, P.; and Caley, R. 1999. *The Festival Text to Speech Synthesis System*.

Chung, G., and Wang, S. S. C. 2003. Automatic acquisition of names using speak and spell mode in spoken dialogue systems. In *Proc. HLT-NAACL*, 197–200.

Cohen, P.; Johnston, M.; McGee, D.; Oviatt, S.; Pittman, J.; Smith, I.; Chen, L.; and Clow, J. 1997. QuickSet: Multimodal interaction for distributed applications. In *Proceedings of the Fifth Annual International Multimodal Conference (Multimedia '97)*, 31–40. ACM Press.

Cohen, W.; Ravikumar, P.; and Fienberg, S. 2003. A comparison of string distance metrics for name-matching task. In *IIWeb Workshop*.

David McKelvie, C. B., and Thompson, H. 1997. Using sgml as a basis for data-intensive nlp. In *Proc. of Applied Natural Language Processing*.

Gallwitz, F.; Noth, E.; and Niemann, H. 1996. A categorical approach for out-of-vocabulary word recognition. In *Proc. of ICSLP '96*.

Glass, J.; Chang, J.; and McCandless, M. 1996. A probabilistic framework for feature-based speech recognition. In *Proc. of ICSLP '96*, 2277–2280.

Gokhan Tur, Robert E. Schapire, D. H.-T. 2003. Active learning for spoken language understanding. In *Proc. of ICASSP-2003*.

Litman, J. H. D. 1999. Prosodic cues to recognition errors. In *Proc.of ASRU*.

Logan, B., and Thong, J. M. V. 2002. Confusion-based query expansion for oov words in spoken document retrieval. In *Proceedings of ICSLP*.

Seneff, S.; Lau, R.; and Polifroni, J. 1999. Organization, communication, and control in the galaxy-ii conversational system. In *Proceedings for Eurospeech '98*.

Strom, N.; Hetherington, L.; Hazen, T.; Sandness, E.; and Glass, J. 1999. Acoustic modeling improvements in a segment-based speech recognizer. In *Proc.of ASRU*.

Walker, M. A. 2000. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. In *Journal of Artificial Intelligence Research*, volume 12.

Yen, K., and Zhao, Y. 1996. Robust speech recognition using a multichannel speech processing front-end. In *Proc.of ICSLP*.