FST-Based Recognition Techniques for Multi-Lingual and Multi-Domain Spontaneous Speech

Timothy J. Hazen, I. Lee Hetherington, and Alex Park

Spoken Language Systems Group
MIT Laboratory for Computer Science
Cambridge, MA, USA
{hazen,ilh,malex}@sls.lcs.mit.edu

Abstract

In this paper we present techniques for building multi-domain and multi-lingual recognizers within a finite-state transducer (FST) framework. The flexibility of the FST approach is also demonstrated on the task of incorporating networks modeling different types of non-speech events into an existing word lattice network. The ability to create robust multi-domain and/or multi-lingual recognizers for spontaneous speech will enable a conversational system to switch seamlessly and automatically among different domains and/or languages. Preliminary results using a bi-domain recognizer exhibit only small recognition accuracy degradation in comparison to domain-dependent recognition. Similarly promising results were observed using a bi-lingual recognizer which performs simultaneous language identification and recognition. When using the FST techniques to add non-speech models to the recognizer, experiments show a 10% reduction in word error rate across all utterances and a 30% reduction on utterances containing non-speech events.

1. Introduction

The primary focus of the Spoken Language Systems Group is the development of conversational systems which can recognize, understand, and respond to spoken requests. Typically these speech systems have been constrained to operate within in a single domain and a single language. These constraints have largely been imposed by the computational restrictions required for real-time processing and by the desire to obtain high accuracy and robustness from the speech recognizer.

The development of the GALAXY Communicator architecture has provided system developers the capability to rapidly develop conversational systems for new domains and/or languages [1]. In our group, we have utilized GALAXY to develop multi-domain systems which incorporate a variety of domains within one large system. Early versions of our multi-domain systems utilized a single multi-domain recognizer and were capable of performing implicit domain switches [2]. At the time, the multi-domain recognizer was not robust enough for general public use. Recently, our multi-domain systems have utilized independent recognizers for each new domain and forced the user to explicitly request a particular domain before asking a query within that domain. In this paper, we present techniques to perform robust and accurate multi-domain (or multi-lingual) speech recognition which will enable our conversational system to implicitly switch between different domains (or languages).

[1]This research was supported by DARPA under Contract N66001-99-1-8904 monitored through the Naval Command, Control, and Ocean Surveillance Center, and by a contract from NTT.

Our recognizer, called SUMMIT, utilizes finite-state transducer (FST) structures to represent all of the individual components utilized within the lexical search. Because the FST representation is based on a solid mathematical foundation, it is easy to create and manipulate the recognizer's search network using basic mathematical functions [3]. Through the use of a few basic FST operations such as composition, concatenation, union, and closure, a wide variety of network topologies can be created in a few simple steps. The development of the FST framework and tools has allowed us to rapidly develop and explore new recognition modeling techniques that would have been tedious or impossible in our older recognition system.

This paper investigates three applications in which different independently created networks are combined within a single network and searched in parallel. In the first application, we create new networks for modeling a variety of different noise and non-speech artifacts into our word lattice network. In the second application, we combine full recognition networks from different domains to create a single multi-domain recognizer. In the final application, we combine recognition networks from different languages to create a single multi-lingual recognizer.

2. Modeling Non-Speech Artifacts

2.1. Building Noise and Non-Speech Artifact Models

In the past, we have not spent much effort worrying about background noises or non-speech artifacts such as coughs or laughs. We have typically relied on a single *trash* model within our recognizer to try to handle all non-speech events. Unfortunately, a single model is not powerful enough to cover the wide range of non-speech events that can occur and words were often inserted by the recognizer in place of the trash model when these events occurred. As has been done previously in many other speech recognition systems, we have added a collection of non-speech models to the recognizer to address this problem [4].

To add non-speech models to our system, we utilize an FST approach first developed for modeling out-of-vocabulary words [5]. We define a set of acoustic models and a network topology for each non-speech type. Each network uses a fully connected topology allowing the noise to be represented by any sequence of the acoustic models used for that noise. It is possible to constrain the sequence of acoustic models used with a transition bigram, although in the experiments in this paper all transitions are considered equally likely. Figure 1 shows an example topology for one type of noise which contains four different acoustic models. The network for this noise is then added in parallel to the word network.

Presented at *EUROSPEECH 2001,* September 3-7, 2001, Aalborg, Denmark

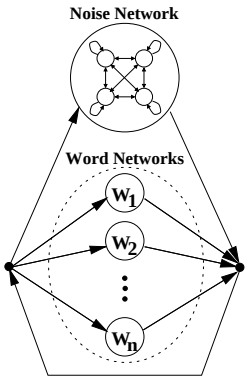

Figure 1: Topology of a finite-state word network with one non-speech (or noise) network added in parallel.

Test Case	Recognizer	# Utts	WER	SER
All data	Baseline	2388	18.9%	37.8%
	+ new models	2388	17.1%	32.7%
Data w/ noise	Baseline	484	64.0%	69.8%
	+ new models	484	45.1%	43.6%
IV Data w/ no noise	Baseline	1716	9.4%	21.9%
	+ new models	1716	9.6%	22.3%
IV Data w/ noise	Baseline	336	46.5%	66.7%
	+ new models	336	28.2%	34.8%

Table 1: Word error rate (WER) and sentence error rate (SER) of the JUPITER recognizer on various test sets when using the baseline recognizer vs. the recognizer containing new models.

2.2. Experimental Results

We evaluate the use of the non-speech models on a test set of 2388 utterances from randomly selected calls made to the JUPITER weather information system [6]. This recognizer has a vocabulary of just over 2000 words and is designed to handle fairly unconstrained queries in the weather domain. The full details of the JUPITER recognizer can be found in [7].

To help improve recognition robustness, we have created five new non-speech model networks to represent coughs, laughs, hang-ups, foreground noises, and background noises. These events are annotated in the training data and are treated as words by the search and language modeling components. An additional weight is applied when entering each non-speech network to regulate its insertion/deletion behavior. This weight is optimized empirically on development data. The acoustic models for each network are seeded from existing models and then retrained iteratively in an unsupervised fashion. The number of different acoustic models used for each non-speech network was determined manually and varied between three and six.

Table 1 shows recognition results under several test cases when using and not using the new models. In the first test case using all 2388 test utterances, the recognizer using the non-speech models exhibited a 10% relative reduction in word error rate from the baseline recognizer. When testing on only the 484 utterances which actually contain non-speech events, a 30% reduction in word error rate is observed. To remove issues caused by the presence of out-of-vocabulary words, the recognizer was also tested on the in-vocabulary portion of the data set. On the 336 in-vocabulary (IV) utterances containing noise, the new models reduced the WER by 39% while degrading the WER on the 1716 IV utterances with no noise by only 2%.

3. Multi-Domain Recognition

3.1. Building Multi-Domain Recognizers

To move from a collection of domain-dependent recognizers towards a single recognizer which can recognize speech from multiple domains, we must address any problems that might arise if the various domain-dependent recognizers were constructed using inherently different modeling approaches. Within our systems, all domain-dependent recognizers utilize the same set of acoustic models trained from pooled data collected from all domains. However, each domain-dependent recognizer could have a different modeling approach in the creation of its lexicon and language model. Though we use class trigram language models for all of our domains, the lexical units and trigram model word classes are often chosen independently for each domain-dependent recognizer.

In this paper we consider two methods for creating a multi-domain recognizer via the combination of domain-dependent recognizers. The first method is to construct a recognizer which allows the networks of each domain-dependent recognizer to be searched in parallel within a single search mechanism. This network of parallel recognizers is easily created within an FST framework by employing the FST union operation. The second approach is to combine and regularize the lexicons, language model classes, and training data from the different domain-dependent recognizers and build a single joint network which can handle all domains.

The parallel network approach has several distinct advantages. First, this approach is extremely easy to implement if the domain-dependent recognizers already exist. After the FST union operation on the set of domain-dependent recognizers is completed, no further work is required and a standard search can be performed on the resulting network. A second advantage is that each domain-dependent recognition network can be constructed and optimized independently of the others. This allows the developer of each domain-dependent recognizer to choose the lexical units and language model classes which are most appropriate for that specific domain independently from the choices made for other domains. A third advantage is that the domain constraint enforced by each domain-dependent network remains intact because movement from one domain-dependent network to another within a single word string hypothesis is prohibited during the search. Because of this constraint, it is also possible to prune away all hypothesized paths within one domain-dependent network if they fail to remain competitive with the top scoring best path from some other domain-dependent network.

The parallel network approach has two potential disadvantages. First, the exact same input/output path could be searched independently within each of the different parallel networks, which could lead to increased computational demands. A second disadvantage is that it may not be possible to compare language model scores from different domain-dependent networks if their inventory of lexical units and language model word classes are drastically different. The incompatibility of scores from different language modeling approaches could result in biases against some domains which might degrade recognition accuracy.

If we utilize the single joint network approach instead, there are two primary advantages; first, the search should be more efficient and, second, the issue of language model incompatibility does not exist. However, there are also several disadvantages to this approach. First, the requirement that the lexicons and language model classes be regularized to one specific form can be

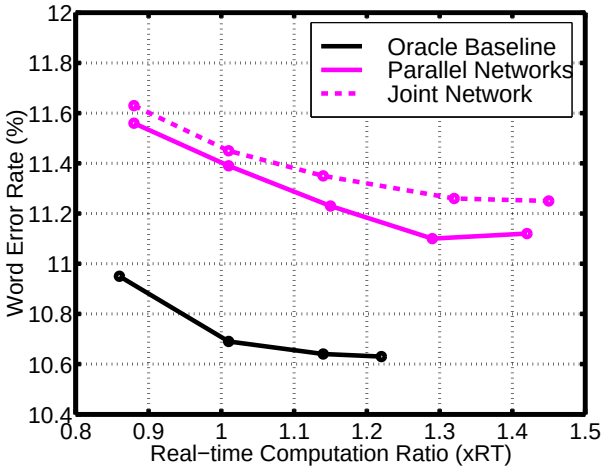

Figure 2: Recognition results on a two domain recognition task using parallel or joint network configurations in comparison to an oracle recognizer which knows the domain *a priori.*

problematic because compromises between different language modeling approaches in different domains must be made, potentially at the expense of recognition accuracy. This regularization process can also be tedious if distinctly different modeling approaches were used for the different domains. Additionally, this recognition approach loses the domain constraint imposed by using the domain-dependent networks in parallel.

3.2. Experimental Results

To investigate the capabilities of a multi-domain recognizer, we combine the recognizers from two domains that we have pursued in our group: weather and air travel. For the weather domain we use the recognizer from the JUPITER weather information system. For air travel we use the recognizer from the MERCURY air travel system [8]. This system is not as mature as our weather system and we have collected far less data in the air travel domain than in the weather domain.

For evaluation, a collection of calls to the weather and air travel systems were randomly used. To separate the issues of recognition in the presence of noise and out-of-vocabulary words from the issues of multi-domain recognition, we only evaluate on the clean, in-vocabulary portion of the test data in this experiment. In total, the test set includes 1716 weather domain utterances and 1087 air travel utterances. The WER on the weather utterances using the JUPITER recognizer is 9.1% under the condition of real-time computation.[2] The WER for the air travel queries using the MERCURY recognizer is 13.8%.

Figure 2 shows the combined recognition performance over all 2803 test utterances under three conditions. The baseline condition is domain-dependent recognition using an oracle to predetermine the domain of the utterance. The second condition is the parallel network recognizer, where the two domain-dependent recognition networks can be searched in parallel. The third condition is the single joint network approach where one recognition network is trained to cover both domains.

Two key points can be drawn from an examination of the results in the figure. First, the parallel network approach slightly out performs the joint network indicating that the domain constraint imposed by the parallel network approach is able to overcome the potential inefficiencies of its search requirements. A second key point is that the parallel network approach has a relative WER degradation from the oracle baseline of only 6.5% (from 10.7% to 11.4%) when running at real time. If the real-time recognition constraint is loosened, a degradation of only 4.7% (from 10.6% to 11.1%) can be realized at 1.3 times real-time. The small degradation in this two domain system implies it may be possible to reliably recognize utterances from more than two domains using only a single recognizer. It is also possible that contextual information, such as the current domain in focus, could be used to improve the recognition performance in an actual conversational system by biasing the recognizer towards the sub-network of the current domain.

[2]Computed using a PC with a 1.5 GHz Pentium 4 processor.

4. Multi-Lingual Recognition

4.1. Building Multi-Lingual Recognizers

One goal in our multi-lingual research efforts is to build systems which perform simultaneous language identification and speech recognition. This approach will allow users to speak to a multi-lingual system without having to explicitly specify what language they wish to speak ahead of time.

Most past research on language identification has focused on the difficult problem of identifying the language of unconstrained spontaneous speech from a moderate number of languages (typically around ten) using only limited amounts of partially-transcribed training data from each language. These approaches typically relied on the phonological constraints of the learned languages to identify the language of test utterances [9, 10, 11].

In our case, we have the advantage of attacking the language identification problem under much easier conditions. For each language in our system we have enough orthographically transcribed data from each language to build an accurate, medium-vocabulary, domain-dependent recognizer. Under these constrained conditions, the most obvious approach to language identification is to simply run multiple recognizers (one for each language) in parallel and choose the language of the recognizer yielding the highest decoding score.

As discussed in the previous section, the FST framework allows multiple recognizers to be combined easily within a single search network. The only significant difference between the multi-domain recognizer from the previous section and the multi-lingual recognizer we wish to construct is that the recognizers for each language use their own set of acoustic models. Under this condition, care must be taken to ensure that the acoustic scores from different acoustic models are comparable. Without some form of acoustic score normalization, language identification could be unfairly biased towards a language whose acoustic models produce higher average raw acoustic scores in general.

In our system, the acoustic model scores are normalized using the following expression:

$$\frac{p(\vec{x}|m)}{\tilde{p}(\vec{x})} \tag{1}$$

In this expression $\vec{x}$ is an acoustic observation, m represents an acoustic model label, and $\tilde{p}(\vec{x})$ is the normalization model [12]. In our case, $\tilde{p}(\vec{x})$ is an approximation of $p(\vec{x})$ and is estimated from the full set of all acoustic models across all languages (with each language receiving equal weighting). This normalization scheme converts the acoustic scores from absolute density scores to relative density scores, hopefully removing any unwanted biases.

4.2. Experiments

To evaluate the multi-lingual recognition approach proposed above, we construct a bilingual recognizer capable of handling weather domain queries in either English or Japanese. For English we use the recognizer from the JUPITER weather information system. For Japanese, we use the recognizer for the Japanese version of JUPITER called MOKUSEI [13].

The same 1716 in-vocabulary weather utterances used earlier are used for our English test utterances. For Japanese, we utilize 1737 in-vocabulary utterances collected from native speakers of Japanese by the MOKUSEI system. As before, we use only in-vocabulary utterances to allow us to focus on the issues of multi-lingual recognition separate from the issues of handling out-of-vocabulary words and severe noises. As mentioned earlier, the English weather domain recognizer achieves an error rate of 9.1% with real-time computation. The Japanese recognizer achieves an error rate of 9.4% with real-time computation.

Figure 3 shows a comparison of recognition results between two different operating conditions. The first condition is language-dependent recognition using an oracle which predetermines the language of the recognizer to be used. The second condition is the bilingual recognizer constructed with the approach discussed above. At an average real-time computation level of 0.9, the bilingual recognizer suffers a relative recognition accuracy degradation of only 4% (from 9.3% to 9.7%). Closer examination shows that English utterances are processed slower (1.06 times real-time) than Japanese utterances (0.77 time real-time). The difference in processing time is due to the fact that the Japanese recognizer has 75% fewer context dependent acoustic models than the English recognizer. The accuracy degradation is caused by the language identification errors made by the bilingual system. In this experiment the language identification error rate was 1.25%. A majority of the language identification errors occurred on utterances containing only one or two words.

5. Discussion & Future Work

In this paper we have demonstrated how different networks can be easily combined in parallel and searched in unison using basic finite-state transducer operations. We have applied this technique to three problems: (1) adding noise models to a word lattice, (2) combining domain-dependent recognition networks to create a multi-domain recognizer, and (3) combining language-dependent recognizers to create a multi-lingual recognizer. In all three cases the experiments the recognizers were created easily and efficiently because of the power of generality provided by the FST framework.

Of particular note, our experiments with bi-domain and bi-lingual recognition showed us that it is was possible to run more than one recognizer in parallel within a single search efficiently and with little recognition accuracy degradation. In future work we will extend this approach to more than two parallel recognizers. While is is clear that each new added domain or language will stress the capabilities of this approach, we believe there is still room for additional parallel networks in both the multi-domain and multi-lingual systems before severe degradation in recognition performance becomes a problem. This will allow us to construct multi-domain (or multi-lingual) conversational systems where the user is not required to specify a domain (or language) before asking a query.

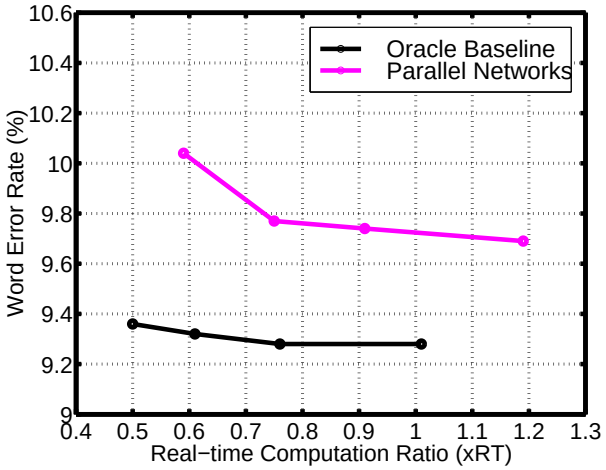

Figure 3: Bilingual (English/Japanese) recognition results using a parallel network configurations in comparison to an oracle recognizer which knows the language *a priori*.

6. Acknowledgements

The authors wish to thank Mikio Nakano and Yasu Minami for their effort developing the Japanese MOKUSEI recognizer, and Issam Bazzi for helping us with FST code that he developed for out-of-vocabulary word modeling.

7. References

[1] S. Seneff, R. Lau, and J. Polifroni, "Organization, communication, and control in the GALAXY-II conversational system," *Eurospeech*, Budapest, September 1999.

[2] D. Goddeau, *at al*, "GALAXY: A human-language interface to on-line travel information," *ICSLP*, Yokohama, September 1994.

[3] F. Pereira and M. Riley, "Speech recognition by composition of weighted finite automata," in *Finite-State Language Processing* (E. Roche and Y. Schabes, eds.), pp. 431–453, Cambridge, MA, MIT Press, 1997.

[4] W. Ward, "Modelling non-verbal sounds for speech recognition," *DARPA Speech and Natural Language Workshop*, Cape Cod, MA, October 1989.

[5] I. Bazzi and J. Glass, "Modeling out-of-vocabulary words for robust speech recognition," *ICSLP*, Beijing, October, 2000.

[6] V. Zue, *et al*, "JUPITER: A telephone-based conversational interface for weather information," *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 1, January 2000.

[7] J. Glass, T. Hazen, and L. Hetherington, "Real-time telephone-based speech recognition in the JUPITER domain," *ICASSP*, Phoenix, March, 1999.

[8] S. Seneff and J. Polifroni, "Dialogue management in the MERCURY flight reservation system," *Satellite Dialogue Workshop, ANLP-NAACL*, Seattle, April, 2000.

[9] M. Zissman, "Comparison of four approaches to automatic language identification of telephone speech" *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, January, 1996.

[10] T. Hazen and V. Zue, "Segment-based automatic language identification," *Journal of the Acoustical Society of America*, vol. 101, no. 4, April, 1997.

[11] J. Navratil and W. Zuehlke, "An efficient phonotactic-acoustic system for language identification," *ICASSP*, Seattle, May, 1998.

[12] S. Kamppari and T. Hazen, "Word and phone level acoustic confidence scoring," *ICASSP*, Istanbul, June, 2000.

[13] V. Zue, *et al*, "From JUPITER to MOKUSEI: Multilingual conversational systems in the weather domain," *MSC2000*, Kyoto, October, 2000.