Stochastic Finite-State models for Spoken Language Machine Translation

Srinivas Bangalore **Giuseppe Riccardi**
AT&T Labs - Research
180 Park Avenue
Florham Park, NJ 07932
{srini,dsp3}@research.att.com

Abstract

Stochastic finite-state models are efficiently learnable from data, effective for decoding and are associated with a calculus for composing models which allows for tight integration of constraints from various levels of language processing. In this paper, we present a method for stochastic finite-state machine translation that is trained automatically from pairs of source and target utterances. We use this method to develop models for English-Japanese and Japanese-English translation. We have embedded the Japanese-English translation system in a call routing task of unconstrained speech utterances. We evaluate the efficacy of the translation system in the context of this application.

1 Introduction

Finite state models have been extensively applied to many aspects of language processing including, speech recognition (Pereira and Riley, 1997; Riccardi et al., 1996), phonology (Kaplan and Kay, 1994), morphology (Koskenniemi, 1984), chunking (Abney, 1991; Srinivas, 1997) and parsing (Roche, 1999). Finite-state models are attractive mechanisms for language processing since they are (a) efficiently learnable from data (b) generally effective for decoding (c) associated with a calculus for composing models which allows for straightforward integration of constraints from various levels of language processing.[1]

In this paper, we develop stochastic finite-state models (SFSM) for statistical machine translation (SMT) and explore the performance limits of such models in the context of translation in limited domains. We are also interested in these models since they allow for a tight integration with a speech recognizer for speech-to-speech translation. In particular we are interested in one-pass decoding and translation of speech as opposed to the more prevalent approach of translation of speech lattices.

The problem of machine translation can be viewed as consisting of two phases: (a) lexical choice phase where appropriate target language lexical items are chosen for each source language lexical item and (b) reordering phase where the chosen target language lexical items are reordered to produce a meaningful target language string. In our approach, we will represent these two phases using stochastic finite-state models which can be composed together to result in a single stochastic finite-state model for SMT. Thus our method can be viewed as a direct translation approach of transducing strings of the source language to strings of the target language. There are other approaches to statistical machine translation where translation is achieved through transduction of source language structure to target language structure (Alshawi et al., 1998b; Wu, 1997). There are also large international multi-site projects such as VERBMOBIL (Verbmobil, 2000) and CSTAR (Woszczyna et al., 1998; Lavie et al., 1999) that are involved in speech-to-speech translation in limited domains. The systems developed in these projects employ various techniques ranging from example-based to interlingua-based translation methods for translation between English, French, German, Italian, Japanese, and Korean.

Finite-state models for SMT have been previously suggested in the literature (Vilar et al., 1999; Knight and Al-Onaizan, 1998). In (Vilar et al., 1999), a deterministic transducer is used to implement an English-Spanish speech translation system. In (Knight and Al-Onaizan, 1998), finite-state machine translation is based on (Brown et al., 1993) and is used for decoding the target language string. However, no experimental results are reported using this approach.

Our approach differs from the previous approaches in both the lexical choice and the reordering phases. Unlike the previous approaches, the lexical choice phase in our approach is decomposed into phrase-level and sentence-level translation models. The phrase-level translation is learned based on joint entropy reduction of the source and target languages and a variable length n-gram model (VNSA) (Riccardi et al., 1995; Riccardi et al., 1996) is learned for the sentence-level translation. For the construc-

[1] Furthermore, software implementing the finite-state calculus is available for research purposes.

tion of the bilingual lexicon needed for lexical choice, we use the alignment algorithm presented in (Alshawi et al., 1998b) which takes advantage of hierarchical decomposition of strings and thus performs a structure-based alignment. In the previous approaches, a bilingual lexicon is constructed using a string-based alignment. Another difference between our approach and the previous approaches is in the reordering of the target language lexical items. In (Knight and Al-Onaizan, 1998), an FSM that represents all strings resulting from the permutations of the lexical items produced by lexical choice is constructed and the most likely translation is retrieved using a target language model. In (Vilar et al., 1999), the lexical items are associated with markers that allow for reconstruction of the target language string. Our reordering step is similar to that proposed in (Knight and Al-Onaizan, 1998) but does not incur the expense of creating a permutation lattice. We use a phrase-based VNSA target language model to retrieve the most likely translation from the lattice.

In addition, we have used the resulting finite-state translation method to implement an English-Japanese speech and text translation system and a Japanese-English text translation system. We present evaluation results for these systems and discuss their limitations. We also evaluate the efficacy of this translation model in the context of a telecom application such as call routing.

The layout of the paper is as follows. In Section 2 we discuss the architecture of the finite-state translation system. We discuss the algorithm for learning lexical and phrasal translation in Section 3. The details of the translation model are presented in Section 4 and our method for reordering the output is presented in Section 5. In Section 6 we discuss the call classification application and present motivations for embedding translation in such an application. In Section 6.1 we present the experiments and evaluation results for the various translation systems on text input.

2 Stochastic Machine Translation

In machine translation, the objective is to map a source symbol sequence $W_S = w_1, \ldots, w_{N_S}$ ($w_i \in L_S$) into a target sequence $W_T = x_1, \ldots, x_{N_T}$ ($x_i \in L_T$). The statistical machine translation approach is based on the *noisy channel* paradigm and the Maximum-A-Posteriori decoding algorithm (Brown et al., 1993). The sequence W_S is thought as a noisy version of W_T and the best guess $\hat{W}_T^*$ is then computed as

$$\hat{W}_T^* = \arg\max_{W_T} P(W_T|W_S)$$

$$= \arg\max_{W_T} P(W_S|W_T)P(W_T) \quad (1)$$

In (Brown et al., 1993) they propose a method for maximizing $P(W_T|W_S)$ by estimating $P(W_T)$ and $P(W_S|W_T)$ and solving the problem in equation 1. Our approach to statistical machine translation differs from the model proposed in (Brown et al., 1993) in that:

- We compute the joint model $P(W_S, W_T)$ from the bilanguage corpus to account for the direct mapping of the source sentence W_S into the target sentence $\hat{W}_T$ that is ordered according to the source language word order. The target string $\hat{W}_T^*$ is then chosen from all possible reorderings[2] of $\hat{W}_T$.

$$\hat{W}_T = \arg\max_{W_T} P(W_S, W_T) \quad (2)$$

$$\hat{W}_T^* = \arg\max_{\hat{W}_T \in \lambda_{W_T}} P(\hat{W}_T \mid \lambda_T) \quad (3)$$

  where λ_T is the target language model and $\lambda_{\hat{W}_T}$ are the different reorderings of $\hat{W}_T$.

- We decompose the translation problem into local (phrase-level) and global (sentence-level) source-target string transduction.

- We automatically learn stochastic automata and transducers to perform the sentence-level and phrase-level translation.

As shown in Figure 1, the stochastic machine translation system consists of two phases, the lexical choice phase and the reordering phase. In the next sections we describe the finite-state machine components and the operation cascade that implements this translation algorithm.

3 Acquiring Lexical Translations

In the problem of speech recognition the alignment between the words and their acoustics is relatively straightforward since the words appear in the same order as their corresponding acoustic events. In contrast, in machine translation, the linear order of words in the source language, in general is not maintained in the target language.

The first stage in the process of bilingual phrase acquisition is obtaining an alignment function that given a pair of source and target language sentences, maps source language word subsequences into target language word subsequences. For this purpose, we use the alignment algorithm described in (Alshawi et

[2]Note that computing the exact set of all possible reorderings is computationally expensive. In Section 5 we discuss an approximation for the set of all possible reorderings that serves for our application.

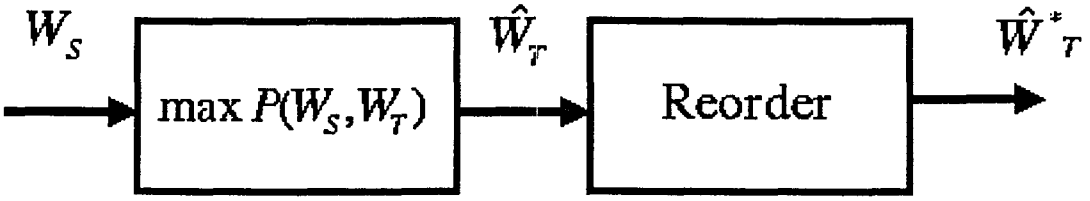

Figure 1: A block diagram of the stochastic machine translation system

English: I need to make a collect call
Japanese: 私は　コレクト　コールを　かける　必要があります
Alignment: 1 5 0 3 0 2 4

English: A T and T calling card
Japanese: エイ　ティー　アンド　ティー コーリング　カード
Alignment: 1 2 3 4 5 6

English: I'd like to charge this to my home phone
Japanese: 私は　これを　私の　家の　電話に　チャージ　したいのです
Alignment: 1 7 0 6 2 0 3 4 5

Table 1: Example bitexts and with alignment information

al., 1998a). The result of the alignment procedure is shown in Table 1.[3]

Although the search for bilingual phrases of length more than two words can be incorporated in a straight-forward manner in the alignment module, we find that doing so is computationally prohibitive.

We first transform the output of the alignment into a representation conducive for further manipulation. We call this a bilanguage $\mathcal{T}_B$. A string $R \in \mathcal{T}_B$ is represented as follows:

$$\mathbf{R} = w_1\_x_1, w_2\_x_2, \ldots, w_N\_x_N \qquad (4)$$

where $w_i \in L_S \cup \epsilon$, $x_i \in L_T \cup \epsilon$, ϵ is the empty string and $w_i\_x_i$ is the symbol pair (colons are the delimiters) drawn from the source and target language.

A string in a bilanguage corpus consists of sequences of tokens where each token $(w_i\_x_i)$ is represented with two components: a source word (possibly an empty word) as the first component and the target word (possibly an empty word) that is the translation of the source word as the second component. Note that the tokens of a bilanguage could be either ordered according to the word order of the source language or ordered according to the word order of the target language. Thus an alignment of a pair of source and target language sentences will result in two bilanguage strings. Table 2 shows an example alignment and the source-word-ordered bilanguage strings corresponding to the alignment shown in Table 1.

Having transformed the alignment for each sentence pair into a bilanguage string (source word-ordered or target word-ordered), we proceed to segment the corpus into bilingual phrases which can be acquired from the corpus $\mathcal{T}_B$ by minimizing the joint entropy $H(L_S, L_T) \simeq -1/M \log P(\mathcal{T}_B)$. The probability $P(W_S, W_T) = P(\mathbf{R})$ is computed in the same way as n-gram model:

$$P(\mathbf{R}) = \prod_i P(w_i\_x_i | w_{i-n+1}\_x_{i-n+1}, \ldots, w_{i-1}\_x_{i-1}) \qquad (5)$$

Using the phrase segmented corpus, we construct a phrase-based variable n-gram translation model as discussed in the following section.

4 Learning Phrase-based Variable N-gram Translation Models

Our approach to stochastic language modeling is based on the Variable Ngram Stochastic Automaton (VNSA) representation and learning algorithms introduced in (Riccardi et al., 1995; Riccardi et al., 1996). A VNSA is a non-deterministic Stochastic Finite-State Machine (SFSM) that allows for parsing any possible sequence of words drawn from a given vocabulary $\mathcal{V}$. In its simplest implementation the state q in the VNSA encapsulates the lexical (word sequence) history of a word sequence. Each

[3]The Japanese string was translated and segmented so that a token boundary in Japanese corresponds to some token boundary in English.

I_私は need_必要があります to_%EPS% make_コールを a_%EPS% collect_コレクト call_かける

I'd_私は like_したいのです to_%EPS% charge_チャージ this_これを to_%EPS% my_私の home_家の phone_電話に

A_エイ T_ティー and_アンド T_ティー calling_コーリング card_カード

Table 2: Bilanguage strings resulting from alignments shown in Table 1. (%EPS% represents the null symbol ϵ).

state recognizes a symbol $w_i \in \mathcal{V} \cup \{\epsilon\}$, where ϵ is the empty string. The probability of going from state q_i to q_j (and recognizing the symbol associated to q_j) is given by the state transition probability, $P(q_j|q_i)$. Stochastic finite-state machines represent in a compact way the probability distribution over all possible word sequences. The probability of a word sequence W can be associated to a state sequence $\xi_W^j = q_1, \ldots, q_j$ and to the probability $P(\xi_W^j)$. For a non-deterministic finite-state machine the probability of W is then given by $P(W) = \sum_j P(\xi_W^j)$. Moreover, by appropriately defining the state space to incorporate lexical and extra-lexical information, the VNSA formalism can generate a wide class of probability distribution (i.e., standard word n-gram, class-based, phrase-based, etc.) (Riccardi et al., 1996; Riccardi et al., 1997; Riccardi and Bangalore, 1998). In Fig. 2, we plot a fragment of a VNSA trained with word classes and phrases. State 0 is the initial state and final states are double circled. The ϵ transition from state 0 to state 1 carries the membership probability $P(C)$, where the class C contains the two elements {`collect, calling card`}. The ϵ transition from state 4 to state 6 is a *back-off* transition to a lower order n-gram probability. State 2 carries the information about the phrase `calling card`. The state transition function, the transition probabilities and state space are learned via the self-organizing algorithms presented in (Riccardi et al., 1996).

4.1 Extending VNSAs to Stochastic Transducers

Given the monolingual corpus $\mathcal{T}$, the VNSA learning algorithm provides an automaton that recognizes an input string W ($W \in \mathcal{V}^N$) and computes $P(W) \neq 0$ for each W. Learning VNSAs from the bilingual corpus $\mathcal{T}_B$ leads to the notion of stochastic transducers τ_{ST}. Stochastic transducers $\tau_{ST} : L_S \times L_T \rightarrow [0,1]$ map the string $W_S \in L_S$ into $W_T \in L_T$ and assign a probability to the transduction $W_S \stackrel{\tau_{ST}}{\rightarrow} W_T$. In our case, the VNSA's model will estimate $P(W_S \stackrel{\tau_{ST}}{\rightarrow} W_T) = P(W_S, W_T)$ and the symbol pair $w_i : x_i$ will be associated to each transducer state q with input label w_i and output label x_i. The model τ_{ST} provides a sentence-level transduction from W_S into W_T. The integrated sentence and phrase-level transduction is then trained directly on the phrase-segmented corpus $\mathcal{T}_B^p$ described in section 3.

5 Reordering the output

The stochastic transducers τ_{ST} takes as input a sentence W_S and outputs a set of candidate strings in the target language with source language word order. Recall that the one-to-many mapping comes from the non-determinism of τ_{ST}. The maximization step in equation 2 is carried out with Viterbi algorithm over the hypothesized strings in L_T and $\hat{W}_T$ is selected. The last step to complete the translation process is to apply the monolingual target language model λ_T to re-order the sentence $\hat{W}_T$ to produce $\hat{W}_T^*$. The re-order operation is crucial especially in the case the bilanguage phrases in $\mathcal{T}_B^p$ are not sorted in the target language. For the re-ordering operation, the exact approach would be to search through all possible permutations of the words in $\hat{W}_T$ and select the most likely. However, that operation is computationally very expensive. To overcome this problem, we approximate the set of the permutations with the word lattice $\lambda_{\hat{W}_T}$ representing $(x_1 \mid x_2 \mid \ldots x_N)^N$, where x_i are the words in $\hat{W}_T$. The most likely string $\hat{W}_T^*$ in the word lattice is then decoded as follows:

$$\begin{aligned} \hat{W}_T^* &= \arg\max(\lambda_T \; o \; \lambda_{\hat{W}_T}) \\ &= \arg \max_{\hat{W}_T \in \lambda_{\hat{W}_T}} P(\hat{W}_T \mid \lambda_T) \end{aligned} \tag{6}$$

where o is the *composition* operation defined for weighted finite-state machines (Pereira and Riley, 1997). The complete operation cascade for the machine translation process is shown in Figure 3.

6 Embedding Translation in an Application

In this section, we describe an application in which we have embedded our translation model and present some of the motivations for doing so. The application that we are interested in is a call type classification task called *How May I Help You* (Gorin et al., 1997). The goal is to sufficiently understand

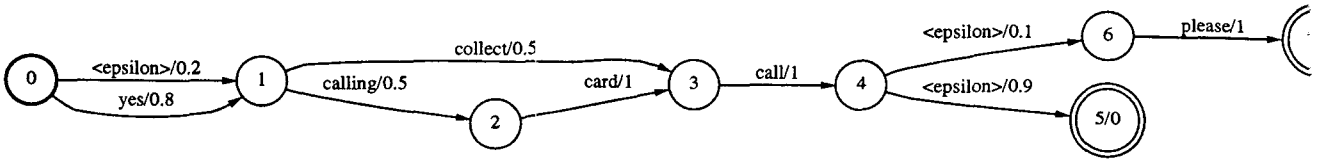

Figure 2: Example of a Variable Ngram Stochastic Automaton (VNSA).

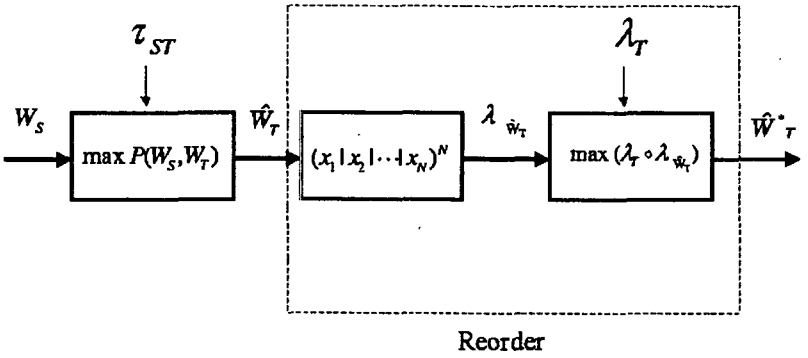

Figure 3: The Machine Translation architecture

caller's responses to the open-ended prompt *How May I Help You?* and route such a call based on the meaning of the response. Thus we aim at extracting a relatively small number of semantic actions from the utterances of a *very large set* of users who are *not trained* to the system's capabilities and limitations.

The first utterance of each transaction has been transcribed and marked with a call-type by labelers. There are 14 call-types plus a class *other* for the complement class. In particular, we focused our study on the classification of the caller's first utterance in these dialogs. The spoken sentences vary widely in duration, with a distribution distinctively skewed around a mean value of 5.3 seconds corresponding to 19 words per utterance. Some examples of the first utterances are given below:

- `Yes ma'am where is area code two zero one?`
- `I'm tryn'a call and I can't get it to go through I wondered if you could try it for me please?`
- `Hello`

We trained a classifer on the training set of English sentences each of which was annotated with a call type. The classifier searches for phrases that are strongly associated with one of the call types (Gorin et al., 1997) and in the test phase the classifier extracts these phrases from the output of the speech recognizer and classifies the user utterance. This is how the system works when the user speaks English.

However, if the user does not speak the language that the classifier is trained on, English, in our case, the system is unusable. We propose to solve this problem by translating the user's utterance, Japanese, in our case, to English. This extends the usability of the system to new user groups.

An alternate approach could be to retrain the classifier on Japanese text. However, this approach would result in replicating the system for each possible input language, a very expensive proposition considering, in general, that the system could have sophisticated natural language understanding and dialog components which would have to be replicated also.

6.1 Experiments and Evaluation

In this section, we discuss issues concerning evaluation of the translation system. The data for the experiments reported in this section were obtained from the customer side of operator-customer conversations, with the customer-care application described above and detailed in (Riccardi and Gorin, January 2000; Gorin et al., 1997). Each of the customer's utterance transcriptions were then manually translated into Japanese. A total of 15,457 English-Japanese sentence pairs was split into 12,204 training sentence pairs and 3,253 test sentence pairs.

The objective of this experiment is to measure the performance of a translation system in the context of an application. In an automated call router there are two important performance measures. The first is the probability of false rejection, where a call is falsely rejected. Since such calls would be transferred to a human agent, this corresponds to a missed opportunity for automation. The second

measure is the probability of correct classification. Errors in this dimension lead to misinterpretations that must be resolved by a dialog manager (Abella and Gorin, 1997).

Using our approach described in the previous sections, we have trained a unigram, bigram and trigram VNSA based translation models with and without phrases. Table 3 shows lexical choice (bag-of-tokens) accuracy for these different translation models measured in terms of recall, precision and F-measure.

In order to measure the effectiveness of our translation models for this task we classify Japanese utterances based on their English translations. Figure 4 plots the false rejection rate against the correct classification rate of the classifier on the English generated by three different Japanese to English translation models for the set of Japanese test sentences. The figure also shows the performance of the classifier using the correct English text as input.

There are a few interesting observations to be made from the Figure 4. Firstly, the task performance on the text data is asymptotically similar to the task performance on the translation output. In other words, the system performance is not significantly affected by the translation process; a Japanese transcription would most often be associated with the same call type after translation as if the original were English. This result is particularly interesting inspite of the impoverished reordering phase of the target language words. We believe that this result is due to the nature of the application where the classifier is mostly relying on the existence of certain key words and phrases, not necessarily in any particular order.

The task performance improved from the unigram-based translation model to phrase unigram-based translation model corresponding to the improvement in the lexical choice accuracy in Table 3. Also, at higher false rejection rates, the task performance is better for trigram-based translation model than the phrase trigram-based translation model since the precision of lexical choice is better than that of the phrase trigram-based model as shown in Table 3. This difference narrows at lower false rejection rate.

We are currently working on evaluating the translation system in an application independent method and developing improved models of reordering needed for better translation system.

7 Conclusion

We have presented an architecture for speech translation in limited domains based on the simple machinery of stochastic finite-state transducers. We have implemented stochastic finite-state models for English-Japanese and Japanese-English translation in limited domains. These models have been trained automatically from source-target utterance pairs. We have evaluated the effectiveness of such a translation model in the context of a call-type classification task.

References

A. Abella and A. L. Gorin. 1997. Generating semantically consistent inputs to a dialog manager. In *Proceedings of European Conference on Speech Communication and Technology*, pages 1879–1882.

Steven Abney. 1991. Parsing by chunks. In Robert Berwick, Steven Abney, and Carol Tenny, editors, *Principle-based parsing*. Kluwer Academic Publishers.

H. Alshawi, S. Bangalore, and S. Douglas. 1998a. Learning Phrase-based Head Transduction Models for Translation of Spoken Utterances. In *The fifth International Conference on Spoken Language Processing (ICSLP98)*, Sydney.

Hiyan Alshawi, Srinivas Bangalore, and Shona Douglas. 1998b. Automatic acquisition of hierarchical transduction models for machine translation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, Montreal, Canada.

P. Brown, S.D. Pietra, V.D. Pietra, and R. Mercer. 1993. The Mathematics of Machine Translation: Parameter Estimation. *Computational Linguistics*, 16(2):263–312.

E. Giachin. 1995. Phrase Bigrams for Continuous Speech Recognition. In *Proceedings of ICASSP*, pages 225–228, Detroit.

A. L. Gorin, G. Riccardi, and J. H Wright. 1997. How May I Help You? *Speech Communication*, 23:113–127.

R.M. Kaplan and M. Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

Kevin Knight and Y. Al-Onaizan. 1998. Translation with finite-state devices. In *Machine translation and the information soup*, Langhorne, PA, October.

K. K. Koskenniemi. 1984. *Two-level morphology: a general computation model for word-form recognition and production*. Ph.D. thesis, University of Helsinki.

Alon Lavie, Lori Levin, Monika Woszczyna, Donna Gates, Marsal Gavalda, , and Alex Waibel. 1999. The janus-iii translation system: Speech-to-speech translation in multiple domains. In *Proceedings of CSTAR Workshop*, Schwetzingen, Germany, September.

Fernando C.N. Pereira and Michael D. Riley. 1997. Speech recognition by composition of weighted finite automata. In E. Roche and Schabes Y.,

Trans VNSA order	Recall (R)	Precision (P)	F-Measure (2*P*R/(P+R))
Unigram	24.5	83.6	37.9
Bigram	55.3	87.3	67.7
Trigram	61.8	86.4	72.1
Phrase Unigram	43.7	80.3	56.6
Phrase Bigram	62.5	86.3	72.5
Phrase Trigram	65.5	85.5	74.2

Table 3: Lexical choice accuracy of the Japanese to English Translation System with and without phrases

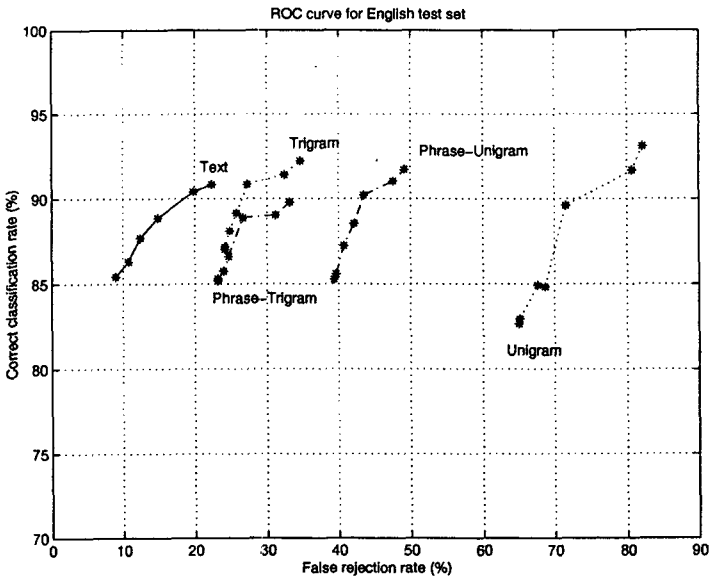

Figure 4: Plots for the false rejection rate against the correct classification rate of the classifier on the English generated by three different Japanese to English translation models

editors, *Finite State Devices for Natural Language Processing*. MIT Press, Cambridge, Massachusetts.

G. Riccardi and S. Bangalore. 1998. Automatic acquisition of phrase grammars for stochastic language modeling. In *Proceedings of ACL Workshop on Very Large Corpora*, pages 188–196, Montreal.

G. Riccardi and A.L. Gorin. January, 2000. Stochastic Language Adaptation over Time and State in Natural Spoken Dialogue Systems. *IEEE Transactions on Speech and Audio*, pages 3–10.

G. Riccardi, E. Bocchieri, and R. Pieraccini. 1995. Non deterministic stochastic language models for speech recognition. In *Proceedings of ICASSP*, pages 247–250, Detroit.

G. Riccardi, R. Pieraccini, and E. Bocchieri. 1996. Stochastic Automata for Language Modeling. *Computer Speech and Language*, 10(4):265–293.

G. Riccardi, A. L. Gorin, A. Ljolje, and M. Riley. 1997. A spoken language system for automated call routing. In *Proceedings of ICASSP*, pages 1143–1146, Munich.

K. Ries, F.D. Buø, and T. Wang. 1995. Improved Language Modeling by Unsupervised Acquisition of Structure. In *Proceedings of ICASSP*, pages 193–196, Detroit.

Emmanuel Roche. 1999. Finite state transducers: parsing free and frozen sentences. In András Kornai, editor, *Extened Finite State Models of Language*. Cambridge University Press.

B. Srinivas. 1997. *Complexity of Lexical Descriptions and its Relevance to Partial Parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, August.

Verbmobil. 2000. Verbmobil Web page. http://verbmobil.dfki.de/.

J. Vilar, V.M. Jiménez, J. Amengual, A. Castellanos, D. Llorens, and E. Vidal. 1999. Text and speech translation by means of subsequential transducers. In András Kornai, editor, *Extened Finite State Models of Language*. Cambridge University Press.

Monika Woszczyna, Matthew Broadhead, Donna Gates, Marsal Gavalda, Alon Lavie, Lori Levin,

and Alex Waibel. 1998. A modular approach to spoken language translation for large domains. In *Proceedings of AMTA-98*, Langhorne, Pennsylvania, October.

Dekai Wu. 1997. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3):377–404.