

# A Finite-State Approach to Machine Translation\*

Srinivas Bangalore and Giuseppe Riccardi

AT&T Labs-Research

180 Park Avenue, Florham Park, NJ 07932.

{srini,dsp3}@research.att.com

## Abstract

The problem of machine translation can be viewed as consisting of two subproblems (a) Lexical Selection and (b) Lexical Reordering. We propose stochastic finite-state models for these two subproblems in this paper. 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. We present a method for learning stochastic finite-state models for lexical choice and lexical reordering that are trained automatically from pairs of source and target utterances. We use this method to develop models for English-Japanese translation and present the performance of these models for translation on speech and text. We also evaluate the efficacy of such a translation model in the context of a call routing task of unconstrained speech utterances.

## 1 Introduction

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) lexical reordering phase where the chosen target language lexical items are rearranged to produce a meaningful target language string. In this paper, we develop stochastic finite-state transducer (SFST) models for these two phases which can then be composed into a single SFST model for Statistical Machine Translation (SMT). We explore the performance limits of such models in the context of translation in limited domains. We are also interested in SFST models since they allow for 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.

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; Bangalore and Joshi, 1999) 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.<sup>1</sup>

A number of approaches to SMT, including the seminal work at IBM (Brown et al., 1993), are stochastic string transductions that map source language strings directly to target language strings. There are other approaches to SMT where translation is achieved through tree transductions that map source language trees to target language trees (Alshawi et al., 1998b; Wu, 1997). There are also international multi-site projects such as VERBMOBIL (VerbMobil, 2000) and CSTAR (Woszczyzna 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.

Unlike previous approaches, we subdivide the translation task into lexical choice and lexical re-

\* Anuvaad is a system embodying our approach and can be seen at <http://www.research.att.com/~srini/Anuvaad.html>

<sup>1</sup>Furthermore, software implementing the finite-state calculus is available for research purposes.

ordering phases. The lexical choice phase is decomposed into phrase-level and sentence-level translation models. We use a tree-based alignment algorithm (Alshawi et al., 1998b) to obtain a bilingual lexicon. The phrase-level translation is learned, based on joint entropy reduction of the source and target languages (Bangalore and Riccardi, 2000). A variable length  $n$ -gram model (VNSA) (Riccardi et al., 1995; Riccardi et al., 1996) is learned for the sentence-level translation. The reordering step uses position markers on a tree-structure, but approximates a tree-transducer using a string-transducer. One of the objectives of this paper is to explore the impact of this approximation on translation accuracy and task accuracy in limited domain applications.

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 algorithms for lexical choice and phrasal translations in Section 3. The details of our method for lexical reordering the result of lexical choice is presented in Section 4. In Section 5 we present the experiments and evaluation results for the various translation systems on text and speech input and in the context of a call-routing spoken dialog system.

## 2 Stochastic Machine Translation

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

$$\begin{aligned} \hat{W}_T^* &= \arg \max_{W_T} P(W_T | W_S) \\ &= \arg \max_{W_T} P(W_S | W_T) P(W_T) \end{aligned} \quad (1)$$

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  $\tilde{W}_T$  that is ordered according to the source language word order. The target string  $\hat{W}_T^*$  is then computed as the most likely string based on the target language model ( $\lambda_T$ ) from a subset of all possible reorderings ( $\lambda_{\tilde{W}_T}$ ) of the string  $\tilde{W}_T$  according to Equation (3).

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

$$\hat{W}_T^* = \arg \max_{\tilde{W}_T \in \lambda_{\tilde{W}_T}} P_{\lambda_T}(\tilde{W}_T) \quad (3)$$

- 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.

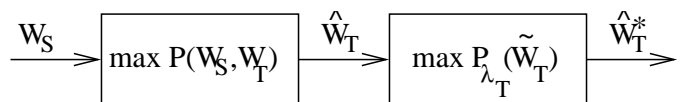


Figure 1: A block diagram of the stochastic machine translation system

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 Lexical Choice

The first stage in the process of training a lexical choice model 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 al., 1998a) which we briefly present here.

The algorithm takes as input a set of bitexts. We define a bitext to be a source language sentence paired with its translation. The algorithm consists of two phases: acquisition of a translation lexicon and an alignment search. The translation lexicon specifies a cost for each pairing of source and target word subsequences<sup>2</sup>. In the second phase, an alignment search is performed that given a source and target sentence pair, produces a set of pairings of

<sup>2</sup>We consider source and target word subsequences of 2-1, 1-2, 1-0 and 0-1

Eng: I need to make a collect call  
 Jap: 私は コレクト コールを かける 必要があります  
 Source: -1 1 4 7 6 4 2  
 Alignment: 1 5 0 3 0 2 4  
 Target: 1 3 4 5 1

Eng: I'd like to charge this to my home phone  
 Jap: 私は これを 私の 家の 電話に チャージ したいのです  
 Source: -1 1 4 2 4 7 5 7 8  
 Alignment: 1 7 0 6 2 0 3 4 5  
 Target: -1 6 2 3 4 7 1

Figure 2: Example bilingual texts with alignment information

minimum total cost which maps the source sentence to its target sentence. This search is carried out in a hierarchical fashion with recursive decomposition of the source and target strings around a hypothesized *head* word in the source string and its corresponding translation in the target string. The hierarchical alignment which minimizes the cost function is computed using a dynamic programming procedure. Some example bitexts and the result of the alignment procedure are shown in Figure 2.<sup>3</sup> The alignment for the first bitext reads as: first source word is aligned to the first target word, the second source word is aligned to the fifth target word, the third source word not aligned with any target word and so on. The tree structure resulting from the hierarchical decomposition of the source string and the target string is represented along the third and the fifth line of Figure 2. Each word position is associated with the word index of its mother in the tree. The root of the tree is indicated by -1. The tree structure information is used for lexical reordering as discussed in Section 4.

Note that we use a tree-based alignment unlike the string-based alignment in IBM statistical models. We believe that a tree-based alignment is more natural for modeling lexical reordering operations than a string-based alignment. We are currently investigating the quality of the dictionary produced by a tree-based alignment compared to a string-based alignment.

From the alignment information in Figure 2, it is straightforward to compile a bilanguage corpus consisting of source-target symbol pair sequences  $\mathcal{T} = \dots(w_i, x_i)\dots$ , where the source word  $w_i \in L_S \cup \epsilon$  and its aligned word  $x_i \in L_T \cup \epsilon$  ( $\epsilon$  is the null symbol). 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. From the cor-

pus  $\mathcal{T}$ , we train a Stochastic Finite State Transducer (SFST) which is an extension of the Variable Ngram State Automaton (Riccardi et al., 1996). 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  $W_S \xrightarrow{\tau_{ST}} W_T$ . In our case, the SFST model will estimate  $P(W_S \xrightarrow{\tau_{ST}} W_T) = P(W_S, W_T)$  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  $\tau_{ST}$  provides a string-to-string transduction from  $W_S$  into  $W_T$ .

### 3.1 Acquiring Phrasal Translations

While word-to-word translation is only approximating the lexical choice process, phrase-to-phrase mapping can greatly improve the translation of collocations, recurrent strings, etc. Moreover, SFSTs can take advantage of the phrasal correlation to improve the computation of the probability  $P(W_S, W_T)$  (Bangalore and Riccardi, 2000). In this section, we describe an alternate method that uses the result of the alignment module as a seed to acquire bilingual phrases of more than two words length.

As mentioned above, we use the alignment information to construct a bilanguage corpus where each token is of the form  $(w_i, x_i)$ . Bilingual phrases can be derived from the phrases (substrings) of the bilanguage corpus that have high mutual information score. We acquire bilanguage phrases from the bilanguage corpus by computing weighted mutual information metric of  $n$ -grams for arbitrarily large values of  $n$ . We use a suffix array to compute the frequencies of large  $n$ -grams similar to the method presented in (Yamamoto and Church, 1998). Since the phrases acquired from a source(target) ordered bilanguage corpus may not have the target(source) language words in the order of the target(source) language, we introduce a reordering phase for the words in a phrase which we call local reordering.

In the local reordering phase, for each phrase we select an alignment which aligns each source word

<sup>3</sup>The Japanese string was translated and segmented so that a token boundary in Japanese corresponds to some token boundary in English.

with some word(s) in the target phrase. We then reorder the words of the target phrase such that the reordering corresponds to a substring (consecutive words) of the target sentence in the selected alignment. A sample set of phrases after reordering is illustrated in Table 3.

Japanese Phrases	English Phrases
エイ ティー アンド ティー	A T and T
私の 家の 電話に	to my home phone
私は コレクト コールを かける 必要があります	I need to make a collect call
私は あなたを どうやって お手伝いしましょうか	how may I help you
はい あなたは いただけますか	yes could you

Figure 3: Examples of acquired phrases after reordering of Japanese phrases

## 4 Lexical Reordering

The lexical choice model outputs a sequence of target language words and phrases for a given source language sentence. Since these target language words and phrases may not form a well-formed target language sentence, we need to apply a lexical reordering (sentence-level) operation.

For the lexical reordering operation, the exact approach would be to search through all possible permutation sequences of words and phrases and select the most likely sequence. However, that is computationally very expensive. To overcome this problem, we decompose the sequence of words and phrases into a tree with each arc labeled with position information of the daughter with respect to its mother. This tree structure could be interpreted as a dependency tree.

We use a stochastic finite-state model to parse the sequence of words and phrases into a tree containing reordering information. We train this SFST from a corpus derived from an aligned corpus of source-ordered target language sentence paired with its target sentence (Figure 4). The corpus (Figure 5) consists of bracketed representation of dependency trees which are constructed from the alignment information shown in Figure 4.

The composition of the reordering finite-state transducer on the result of the lexical choice model results in strings that are annotated with reordering instructions. To ensure we obtain well-formed bracketed strings, we compose the result with a transducer that checks for all possible well-formed brackets, for a fixed number of brackets. This can be regarded as a finite-state approximation of a parathesis context-free grammar upto a bounded depth. The resulting string from the composition contains reordering instructions which are interpreted to form

the reordered target language sentence. Other interesting approaches involve extracting a context-free grammar from the training corpus and approximating the resulting grammar by a finite-state grammar using techniques discussed in (Pereira and Wright, 1997; Nederhof, 2000).

Figure 6 shows the sequence of transductions starting from a source language string that results in a target language string. The intermediate steps involved include lexical choice, parse of the source-ordered target string, reordered parse tree for the target string and the final target string.

## 5 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 in (Riccardi and Gorin, 2000). 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.

### 5.1 Evaluation of Machine Translation Systems

Evaluation of a machine translation systems has been a subject of discussion for many years (Council, 1966; Arnold, 1993). A universally acceptable, objective and reliable metric that can be computed automatically is yet to be found. However, in the interest of evaluating our translation system automatically and objectively without human intervention, we report the performance of a machine translation system in application independent and in the context of an application.

For the application independent evaluation, we employ two metrics based on string edit distance between the output of a translation system and the reference translation string: simple accuracy and translation accuracy (Alshawi et al., 1998b). Simple accuracy is the number of insertion ( $I + I'$ ), deletion ( $D + D'$ ) and substitutions ( $S$ ) errors between the target language strings in the test corpus and the strings produced by the translation model. The metric is summarized in Equation 4.  $R$  is the number of tokens in the target string. This metric is similar to the string distance metric used for measuring speech recognition accuracy.

$$Simple Accuracy = \left(1 - \frac{I + I' + D + D' + S}{R}\right) \quad (4)$$

The simple accuracy metric, however, penalizes a misplaced token twice, as a deletion from its expected position and insertion at a different posi-

Eng-Jap: 私は したいのです チャージ これを 私の 家の 電話に  
 Japanese: 私は これを 私の 家の 電話に チャージ したいのです  
 Source: -1 4 2 1 6 4 6  
 Alignment: 1 7 6 2 3 4 5  
 Target:-1 1 4 2 4 7 2

Figure 4: Alignment between English-ordered Japanese and Japanese strings

私は:私は     $\epsilon$ :+1  
 $\epsilon$ : [  
 $\epsilon$ : [  したいのです:したいのです     $\epsilon$ :-1     $\epsilon$ : [  チャージ:チャージ     $\epsilon$ : ]     $\epsilon$ : ]  
 $\epsilon$ :+2  これを:これを     $\epsilon$ :+1  
 $\epsilon$ : [     $\epsilon$ : [ 私の:私の     $\epsilon$ : ]     $\epsilon$ :-1  家の:家の     $\epsilon$ :+1     $\epsilon$ : [  電話に:電話に     $\epsilon$ : ]     $\epsilon$ : ]  
 $\epsilon$ : ]

Figure 5: Bracket representation of a dependency tree with information on reordering words. Each token consists of the form of a transduction (input:output).

tion. We use a second metric, Translation Accuracy, shown in Equation 5, which treats deletion of a token at one location in the string and the insertion of the same token at another location in the string as one single movement error ( $M=I+D$ ).<sup>4</sup> This is in addition to the remaining insertion, deletion and substitutions.

$$Translation Accuracy = \left(1 - \frac{M + I' + D' + S}{R}\right) \quad (5)$$

For application dependent evaluation of a translation system, we employ the translation system in the context of call type classification. We compare the classification accuracy using the text produced by the translation system against that produced using the reference text.

## 5.2 Application Independent Evaluation

Using the training sentence pairs and the procedure described in the earlier sections, we have developed English to Japanese and Japanese to English translation systems.

Table 1 presents the performance results of the English to Japanese translation system using different translation models, before and after the reordering stage.

In both tables, the unigram, bigram and trigram translation models do not include any phrases while uniphase, biphrase and triphrase models include the automatically acquired phrases. As can be seen, the performance of models after reordering is significantly better than the performance before reordering.

<sup>4</sup>Note that the movement errors are derived after the strings are compared using insertion, deletion and substitution operations.

Trans VNSA order	Accuracy before Reordering	Accuracy after Reordering
Unigram	23.8	32.2
Bigram	56.9	69.4
Trigram	56.4	69.1
UniPhrase	44.0	46.8
BiPhrase	60.4	69.8
TriPhrase	58.9	66.7

Table 1: Translation Accuracy of the English to Japanese Translation System with and without phrases, before and after reordering on text.

### 5.2.1 Spoken Language Translation

The English-Japanese translation system was used to translate spoken language as well. The composed lexical choice transducer and lexical reordering transducer can be directly plugged into a speech recognizer in conjunction with the source language acoustic model to produce a source-speech to target-text system. We will report the result of such a system in the final version of this paper. Currently, we report performance on one-best output of a speech recognizer as the input to the translation system.

A VNSA-based trigram language model that was trained on the 12204 training sentences was used as the language model for the speech recognizer. An off-the-shelf context dependent acoustic model for telephone speech was used as the acoustic model. The word accuracy of the speech recognizer on the test data is 74.3%. Table 2 summarizes the translation accuracies of various models on the one-best output of the speech recognizer. The simple and translation accuracy of the triphrase-based translation system on the one-best output of the recognizer is 56.9% respectively.

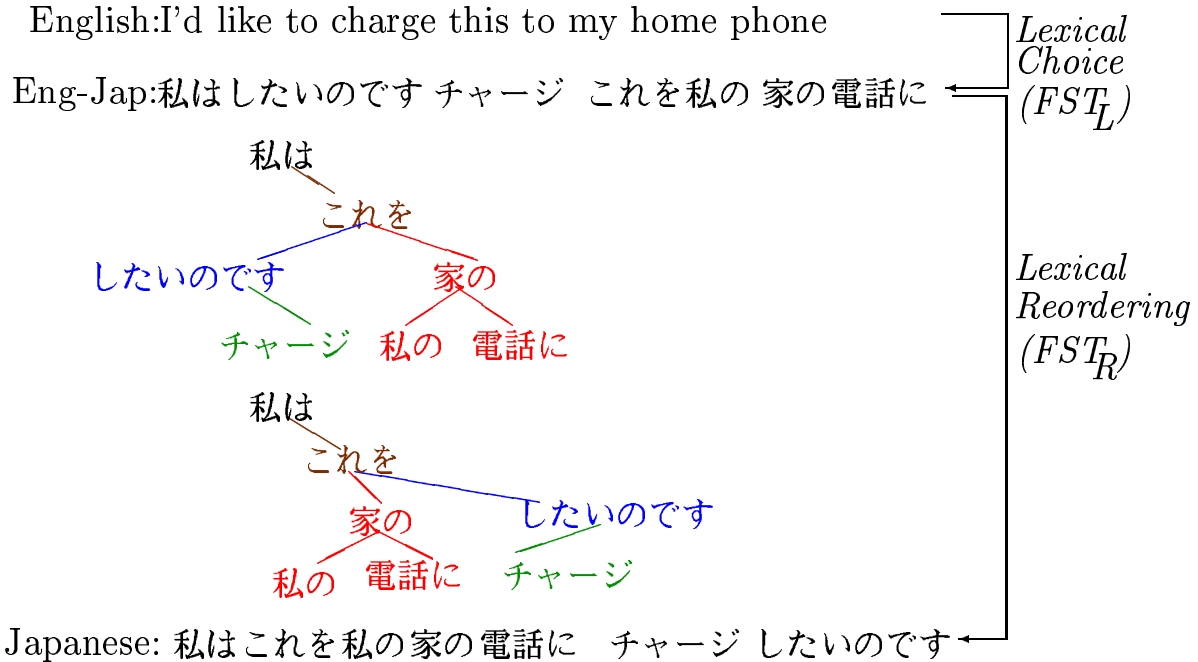


Figure 6: Sequence of finite-state transductions from English to Japanese

Trans VNSA order	Accuracy before Reordering	Accuracy after Reordering
Unigram	21.4	21.7
Bigram	48.9	55.7
Trigram	49.0	56.8
UniPhrase	39.3	39.6
BiPhrase	51.3	56.5
TriPhrase	50.9	56.9

Table 2: Translation Accuracy of the English to Japanese Translation System with and without phrases, before and after reordering on one-best output of the speech recognizer.

### 5.3 Application Dependent Evaluation: Call Type Classification

The objective of this experiment is to measure the performance of a translation system in the context of an application, in our case, a call type classification application task called the *How May I Help You?* (Gorin et al., 1997) task. We briefly review the problem and the spoken language system. The goal is to sufficiently understand 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 try'n'a call and I can't get it to go through I wondered if you could try it for me please?
- Hello

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 or classified as *other*. 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.

Trans VNSA order	Recall ( $R$ )	Precision ( $P$ )	F-Measure $\frac{2*P*R}{P+R}$
Unigram	31.1	92.2	46.5
Bigram	65.4	89.9	75.8
Trigram	63.2	91.5	74.7
Phr. Unigram	41.9	92.9	57.8
Phr. Bigram	66.7	89.3	76.4
Phr. Trigram	65.3	89.9	75.7

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

In order to measure the effectiveness of our translation models for this task we classify Japanese utterances based on their English translations. We trained a classifier 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 translation output. Figure 7 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.

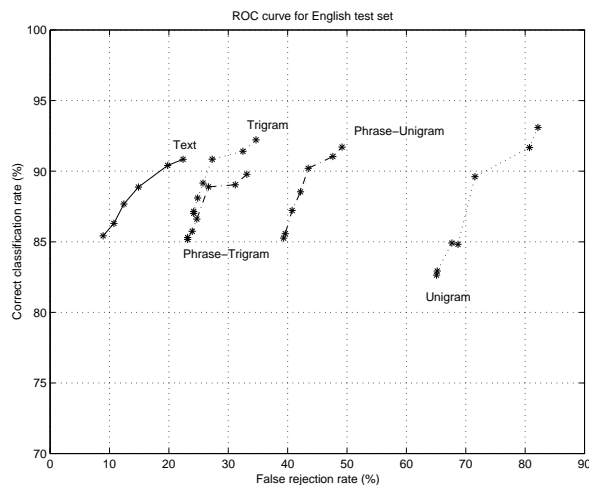


Figure 7: Plots for the false rejection rate against the correct classification rate of the classifier on the English generated by three different Japanese to English translation models

There are a few interesting observations to be made from the Figure 7. Firstly, the task performance on the text data is asymptotically similar to the task performance on the translation out-

put. 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. 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.

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.

## 6 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 36<sup>th</sup> Annual Meeting of the Association for Computational Linguistics*, Montreal, Canada.

- E. Arnold. 1993. Special issue on evaluation of MT Systems. *Machine Translation*, 8(1-2):1–126.
- Srinivas Bangalore and Aravind Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2).
- Srinivas Bangalore and Giuseppe Riccardi. 2000. Stochastic finite-state models for spoken language machine translation. In *Proceedings of the Workshop on Embedded Machine Translation Systems*.
- 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.
- National Research Council. 1966. Alpac report. Technical report, National Academy of Sciences.
- 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 Wozszczyna, Donna Gates, Marsal Gavaldà, , and Alex Waibel. 1999. The janus-iii translation system: Speech-to-speech translation in multiple domains. In *Proceedings of CSTAR Workshop*, Schwetzingen, Germany, September.
- Mark-Jan Nederhof. 2000. Practical experiments with regular approximation of context-free languages. *Computational Linguistics*, 26(1).
- Fernando C.N. Pereira and Michael D. Riley. 1997. Speech recognition by composition of weighted finite automata. In E. Roche and Schabes Y., editors, *Finite State Devices for Natural Language Processing*. MIT Press, Cambridge, Massachusetts.
- Fernando C.N Pereira and Rebecca Wright. 1997. Finite-state approximation of phrase-structure grammars. In E. Roche and Y. Schabes, editors, *Finite-State Language Processing*. MIT Press.
- G. Riccardi and A.L. Gorin. 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.
- Emmanuel Roche. 1999. Finite state transducers: parsing free and frozen sentences. In András Kornai, editor, *Extended Finite State Models of Language*. Cambridge University Press.
- 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, *Extended Finite State Models of Language*. Cambridge University Press.
- Monika Wozszczyna, Matthew Broadhead, Donna Gates, Marsal Gavaldà, 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.
- M. Yamamoto and K. W. Church. 1998. Using Suffix Arrays to Compute Term Frequency and Document Frequency for All Substrings in a Corpus. In *Proceedings of ACL Workshop on Very Large Corpora*, pages 28–37, Montreal.