
Machine Translation **17:** 165–184, 2002.

Stochastic Finite-State Models for Spoken Language Machine Translation

SRINIVAS BANGALORE and GIUSEPPE RICCARDI
AT&T Labs Research, 180 Park Avenue, Florham Park, NJ 07932, USA
E-mail: 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. In this paper, we propose stochastic finite-state models for these two subproblems. 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 selection and lexical reordering that are trained automatically from pairs of source and target utterances. We use this method to develop models for English–Japanese and English–Spanish 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.

Key words: speech-to-speech translation, stochastic finite-state transducers, spoken language dialog systems

1. Introduction

The problem of machine translation (MT) can be viewed as consisting of two subproblems: (a) lexical selection, where appropriate target-language lexical items are chosen for each source-language lexical item and (b) lexical reordering, where the chosen target-language lexical items are rearranged to produce a meaningful target-language string. We have proposed stochastic finite-state transducer (SFST) models for these two subproblems (Bangalore and Riccardi, 2000, 2001) which can then be composed into a single SFST model for Statistical Machine Translation (SMT).[1] 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 recognition and translation of speech as opposed to the more prevalent approach of translation of speech recognition transcriptions.

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 and (c) associated with a calculus for composing models which allows for straightforward integration of constraints from various levels of language processing.[2]

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., 1998a; Wu, 1997). There are also international multi-site projects such as VERBMOBIL (Wahlster, 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 MT 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 selection and lexical reordering subproblems. The lexical selection subproblem is decomposed into phrase-level and sentence-level translation models. We use a tree-based alignment algorithm (Alshawi et al., 1998a) 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, 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. We 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 Japanese–English and Spanish–English text translation systems. 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 mathematical model of the finite-state translation system. We discuss the algorithms for lexical selection and phrasal translations in Section 4. The details of our method for lexical reordering the result of lexical selection is presented in Section 5. In Section 6 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 MT, 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 SMT 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 of as a noisy version of W_T and the best guess $\hat{W}_T^*$ is then computed as (1).

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

Brown et al. (1993) 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 SMT 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 a target sentence ($\hat{W}_T$) that is ordered according to the source language word order (2). The target string $\hat{W}_T^*$ is then computed as the most likely string based on the target language model (λ_T) from a set of possible reorderings ($\lambda_{\hat{W}_T}$) of the string $\hat{W}_T$ according to (3):

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

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

– We decompose the translation problem into local (phrase-level) (2) and global (sentence-level) (3) source–target string transduction.
– We automatically learn stochastic automata and transducers to perform the phrase-level and sentence-level translation.

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

3. Learning Phrase-based Variable n-Gram Translation Models

Our approach to stochastic language modeling is based on the Variable n-gram Stochastic Automaton (VNSA) representation and learning algorithms introduced in Riccardi et al. (1995, 1996). A VNSA is a non-deterministic 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

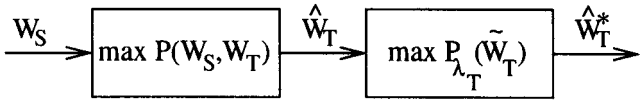

Figure 1. A block diagram of the stochastic MT system.

sequence) history of a word sequence. Each 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, $P(q_j|q_i)$. SFSMs 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 distributions (i.e., standard word n-gram, class-based, phrase-based, etc.) (Riccardi et al., 1996, 1997; Riccardi and Bangalore, 1998). In Figure 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).

3.1. 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 τ_{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 4.1.

4. Lexical Selection

The first stage in the process of training a lexical selection model is obtaining an alignment function that given a pair of source- and target-language sentences, maps

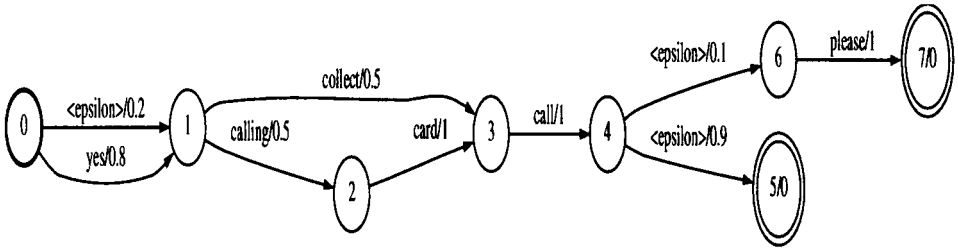

Figure 2. Example of a Variable n-gram Stochastic Automaton (VNSA).

source-language word subsequences into target-language word subsequences. For this purpose, we use the alignment algorithm described in Alshawi et al. (1998b) 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.[3] In the second phase, an alignment search is performed that given a source and target sentence pair, produces a set of pairings of 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. The result of this alignment procedure is the mapping among words of the source language and the target language as well as a dependency tree structure for the source and target language strings. Note that the dependency tree structure might not correspond to the linguistic dependency structure of the sentences since the decomposition of the sentence is primarily driven by the need to minimize swapping of aligned words.

Some example bitexts and the result of the alignment procedure are shown in Figure 3 and graphically depicted in Figure 4.[4] 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 dependency 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 3. 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 dependency tree structure information is used for lexical reordering as discussed in Section 5.

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.

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
Source dependency tree: -1 1 4 7 6 4 2
Alignment: 1 5 0 3 0 2 4
Target dependency tree: -1 3 4 5 1

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
Source dependency tree: -1 1 4 2 4 7 5 7 8
Alignment: 1 7 0 6 2 0 3 4 5
Target dependency tree: -1 6 2 3 4 7 1

Figure 3. Example bilingual texts with alignment information.

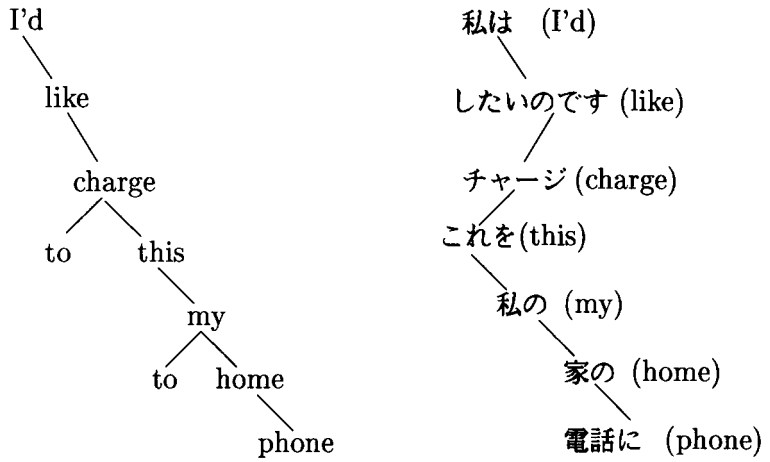

Figure 4. Graphical representation of the information present in the alignment.

From the alignment information in Figure 3, it is straightforward to compile a bilanguage corpus consisting of source–target symbol pair sequences $\mathcal{T} = \ldots(w_i : x_i)\ldots$, where the source word $w_i \in L_S \cup \epsilon$ and its aligned word $x_i \in L_T \cup \epsilon$ (ϵ 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. Figure 5 shows an example alignment and the source-word-ordered bilanguage strings corresponding to the alignment shown in Figure 3. From the corpus $\mathcal{T}$, we train an SFST which is an extension of the VNSA (Riccardi et al., 1996).

I:私は need:必要があります to:ϵ make:コールを
a:ϵ collect_コレクト call_かける

I'd:私は like:したいのです to:ϵ charge:チャージ this:これを
to:ϵ my:私の home:家の phone:電話に

Figure 5. Bilanguage strings resulting from alignments shown in Figure 2.

4.1. ACQUIRING PHRASAL TRANSLATIONS

While word-to-word translation is only approximating the lexical selection 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 $(w_i : x_i)$ (Figure 5). Bilingual phrases can be derived from the phrases (substrings) of the bilinguage 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 ordered bilanguage corpus may not have the target-language words in the order of the target language, we introduce a reordering phase for the words in a phrase which we call "local reordering".

For local reordering, we use the local syntactic constraints encoded in an n-gram target-language model to recover the preferred ordering of the words present in the phrase to be reordered. There are several methods to achieve this result. One such method is to represent the words in the phrase to be reordered as a "sausage" finite-state model as shown in Figure 6. This representation encodes all possible permutations of the words including strings involving repetitions of words.[5] The sausage finite-state model is composed with the n-gram target language and the best reordering is recovered according to equation (4),

$$\text{Locally reordered phrase} = \min(\lambda_s \circ \lambda_{LM}) \tag{4}$$

where the locally reordered phrase has the lowest cost among all the phrases contained in λ_s. The cost is determined by the language model λ_{LM}. The result of reordering the phrases is shown in Figure 7.

The bilanguage corpus is phrase-segmented using the acquired phrases and an n-gram VNSA-based stochastic transducer model is built using this segmented cor-

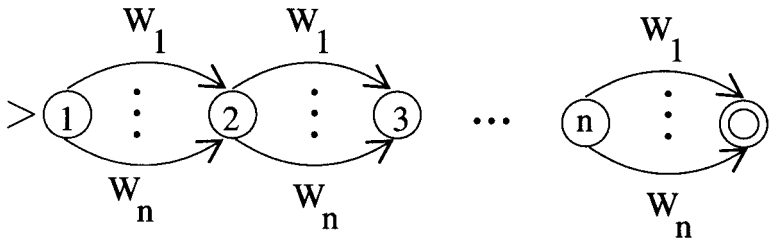

Figure 6. A "sausage" representation for a set of words $\{W_1 \ldots W_n\}$.

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 7. Examples of acquired phrases after reordering of Japanese phrases.

pus. The arcs of the resulting transducer associate the words/phrases of the source language with the words/phrases of the target language. Furthermore, since these associations are made in the n-gram context in the transducer, they encapsulate the local contextual information often needed for lexical selection. It would be interesting to study the influence of longer range contextual information beyond n-grams (such as those contributed from syntax) on lexical selection which are not modeled by the lexical selection transducer.

5. Lexical Reordering

The lexical selection model outputs a sequence of target-language words and phrases for a given source language sentence as shown by the finite-state transducer in Figure 8. Note that the input transitions of the finite-state transducer encode the source-language sentence and the output transitions encode the target-language sentence. In translation among closely related languages such as English–Spanish, the lexical selection model with local reordering might be sufficient to form a well-formed target-language sentence. However, for translation between English–Japanese, the ordering of the target-language words and phrases may not form

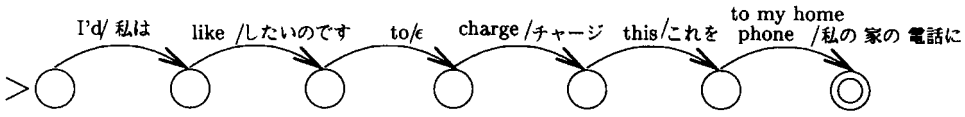

Figure 8. Finite-state transducer representation for lexical selection.

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 9. Alignment between English-ordered Japanese and Japanese strings.

a well-formed target-language sentence. We need to apply a lexical reordering (sentence-level) operation (see equation (3)).

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.

To obtain the dependency tree with reordering information, we reuse the alignment algorithm discussed in Section 4 to align the source-ordered target sentence and the target sentence. Figure 9 shows the English-ordered Japanese string paired with the Japanese string and the result of aligning these two strings. The result of the alignment procedure is shown graphically in Figure 10. We transform the alignment shown in Figure 9 into a corpus (Figure 11) consisting of bracketed representation of dependency trees. The tokens of this bilanguage corpus are tuples either of the form $w_i : w_i$ where $w_i \in L_T$ or the reordering instruction tokens of the form ϵ: x_i where $x_i \in [,], +1, -1, +2, -2$. The corpus is created by an in-order traversal of the English-ordered Japanese dependency tree with brackets to indicate subtrees. Additionally, reordering tokens are inserted in order to indicate the reordering of the subtrees in the source dependency tree with respect to the target dependency tree. The reordering tokens provide linear order information of the children subtrees with respect to its parent node. For example したいのです *shitainodesu* 'like' appears as the second child to the right of its parent これを *sore o* 'this' and hence the subtree rooted in したいのです *shitainodesu* is preceeded with a reordering instruction ϵ: $+2$ in Figure 11. We use the resulting corpus to train a reordering finite-state transducer.

The composition of the reordering finite-state transducer with the result of the lexical selection model results in strings that are annotated with reordering instructions. To ensure we obtain well-formed bracketed strings, we compose the result

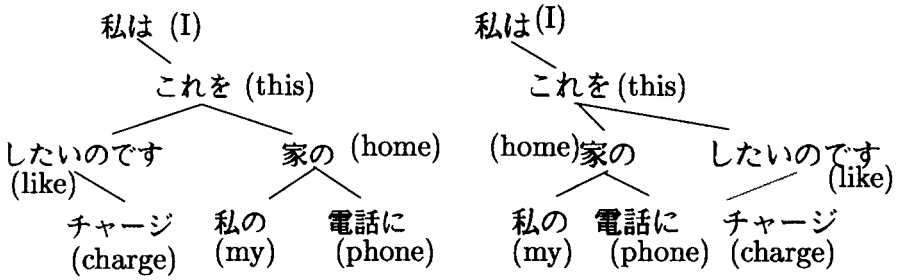

Figure 10. Graphical representation of the information present in the alignment for the English-ordered-Japanese and Japanese sentence.

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

Figure 11. Bracket representation of a dependency tree with information on reordering words. Each token consists of the form of a transduction (input:output).

with a transducer that checks for all possible well-formed brackets, for a fixed number of bracket depth. This can be regarded as a finite-state approximation of a parathensis context-free grammar up to 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 to lexical reordering 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) and Nederhof (2000).

Figure 12 shows the sequence of transductions starting from a source-language string that results in a target-language string. The intermediate steps involved include lexical selection, parse of the source-ordered target string, reordered parse tree for the target string and the final target string $\hat{W}_T^*$ (5),

$$\hat{W}_T^* = \min(W_S \circ \lambda_L \circ \lambda_R) \tag{5}$$

where the source string W_S is composed with the lexical selection transducer (λ_L) and the lexical reordering transducer (λ_R). The min function returns the lowest-cost path in a transducer.

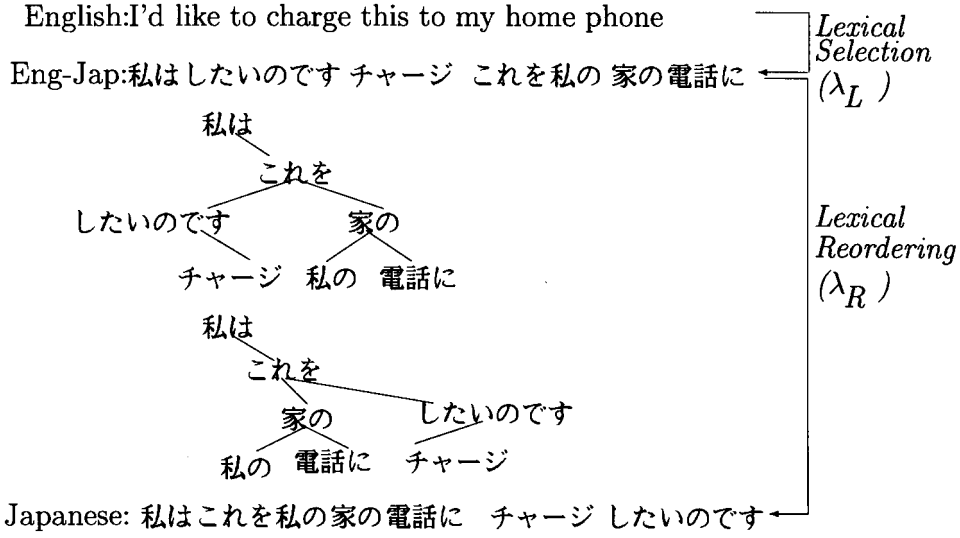

Figure 12. Sequence of finite-state transductions from English to Japanese.

6. 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 call routing application described in Riccardi and Gorin (2000). Each of the customer's utterance transcriptions was then manually translated into Japanese and Spanish. A total of 15,457 English–Japanese and English–Spanish sentence pairs was split into 12,204 training sentence pairs and 3,253 test sentence pairs.

6.1. EVALUATION OF MACHINE TRANSLATION SYSTEMS

Evaluation of MT systems has been a subject of discussion for many years (ALPAC, 1966; Arnold et al., 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 an MT system both 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., 1998a). Simple accuracy is the number of insertion ($I = I' + I''$), deletion ($D = D' + D''$) and substitution (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 (6). 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.

$$\text{Simple Accuracy} = (1 - \frac{I + D + S}{R}) \times 100 \tag{6}$$

The simple accuracy metric, however, penalizes a misplaced token twice, as a deletion from its expected position and insertion at a different position. We use a second metric, Translation Accuracy, shown in equation (7), 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 ($M = I' + D'$).[6] This is in addition to the remaining insertions, deletions and substitutions.

$$\text{Translation Accuracy} = (1 - \frac{M + I'' + D'' + S}{R}) \times 100 \tag{7}$$

An alternate evaluation metric has been proposed in recent literature (Papineni et al., 2002), which relaxes the string comparison metric and is based on the number of overlapping n-grams between several source translations and the system-generated translation.

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.

6.2. APPLICATION-INDEPENDENT EVALUATION

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

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

In both tables, the unigram, bigram and trigram translation models do not include any phrases while uniphrase, 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.

6.2.1. *Spoken Language Translation*

The English–Japanese translation system was used to translate spoken language as well. The composed lexical selection 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.

Table I. Translation accuracy of the English–Japanese translation system with and without phrases, before and after reordering on text.

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

A VNSA-based trigram language model that was trained on the 12,204 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%.[7] Table II summarizes the translation accuracies of various models on the one-best output of the speech recognizer. The translation accuracy of the triphrase-based translation system on the one-best output of the recognizer is 56.9%. The decoding network for the speech recognition and translation decoder is computed as in (8),

$$\lambda_C \circ \lambda_L \circ \lambda_L \circ \lambda_R \tag{8}$$

where λ_C and λ_L are the transducers encoding the acoustic and lexicon models (Pereira and Riley, 1997; Riccardi et al., 1996).

6.2.2. *Lexical Selection Accuracy*

Using our approach described in the previous sections, we have trained a unigram, bigram and trigram VNSA-based Japanese–English translation models with and without phrases. Table III shows lexical selection accuracy for these different translation models measured in terms of recall, precision and F-measure. If *Ref* is the set of words in the reference translation and *Res* is the set of words in the translation output, then the metrics are computed as in (9)–(11).

$$\text{Recall} = \left(\frac{| \, Res \cap Ref \, |}{| \, Ref \, |} \right) \times 100 \tag{9}$$

$$\text{Precision} = \left(\frac{| \, Res \cap Ref \, |}{| \, Res \, |} \right) \times 100 \tag{10}$$

Table II. Translation accuracy of the English–Japanese translation system with and without phrases, before and after reordering on one-best output of the speech recognizer.

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 III. Lexical selection accuracy of the Japanese–English translation system with and without phrases.

Trans VNSA order	Recall (R)	Precision (P)	F-Measure $\frac{2\times P\times R}{(P+R)}$
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

$$F - \text{measure} = \left(\frac{2 \times \text{Precision} \times \text{Recall}}{(\text{Precision} + \text{Recall})}\right) \times 100 \tag{11}$$

The accuracy of lexical selection plays an important role in classification of utterances, an application we discuss in the following section.

6.3. APPLICATION-DEPENDENT EVALUATION: CALL ROUTING

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 routing 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 understand sufficiently 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 (such as *BILLING_CREDIT*, *CALLING_CARD*, *AREA_CODE*, *DIAL_FOR_ME* etc.) 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 in (12).

(12) a. Yes ma'am where is area code two zero one?

b. I'm tryn'a call and I can't get it to go through I wondered if you could try it for me please?

c. 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).

There has been extensive effort in developing this call routing application for English. The dialog flow and the domain semantics have been carefully crafted for this application. In order to make this application multilingual, we propose embedding an MT component as a front-end to this application. Figure 13 illustrates this idea. The translation component would translate non-English user utterances to English and translate the system's English utterances into the user's language. The goal is to avoid the extensive development effort for building this application for each new language.

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 14 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.

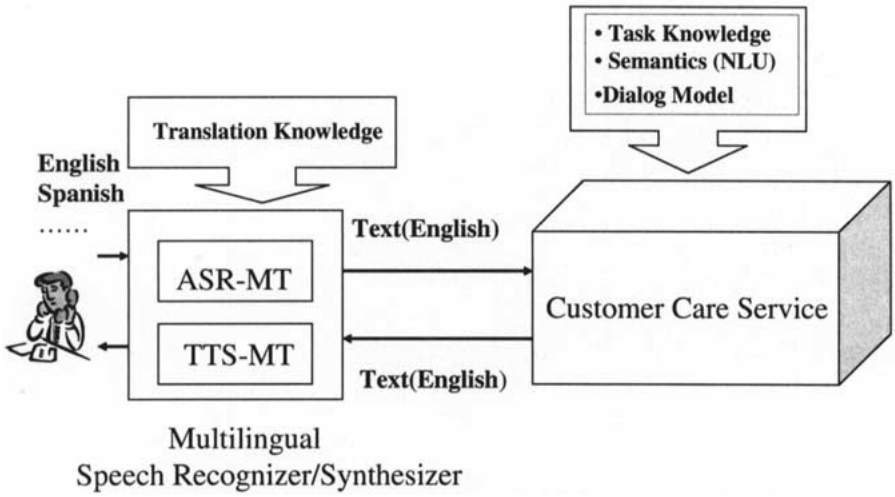

Figure 13. Multilingual-enabling an existing monolingual dialog application.

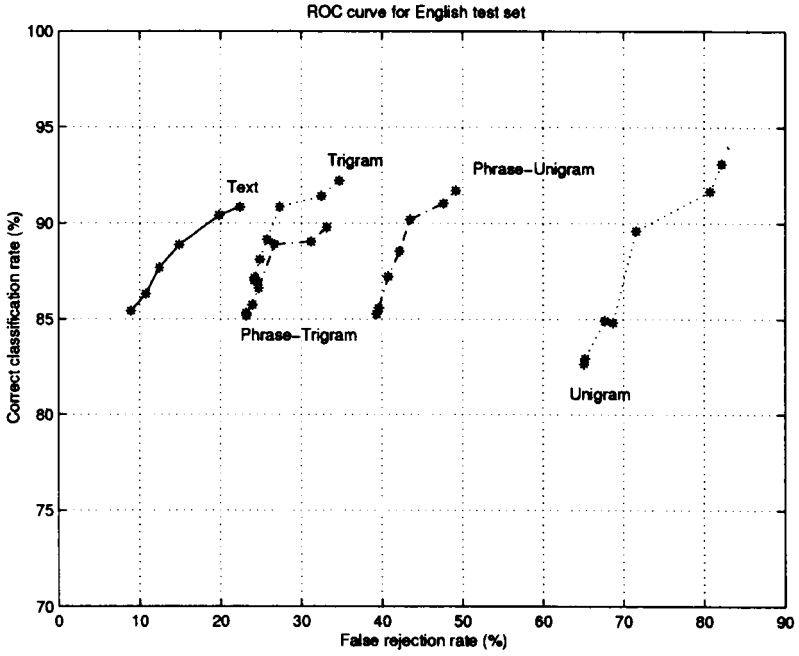

Figure 14. Plots for the false rejection rate against the correct classification rate of the classifier on the English generated by three different Japanese-to-English translation models.

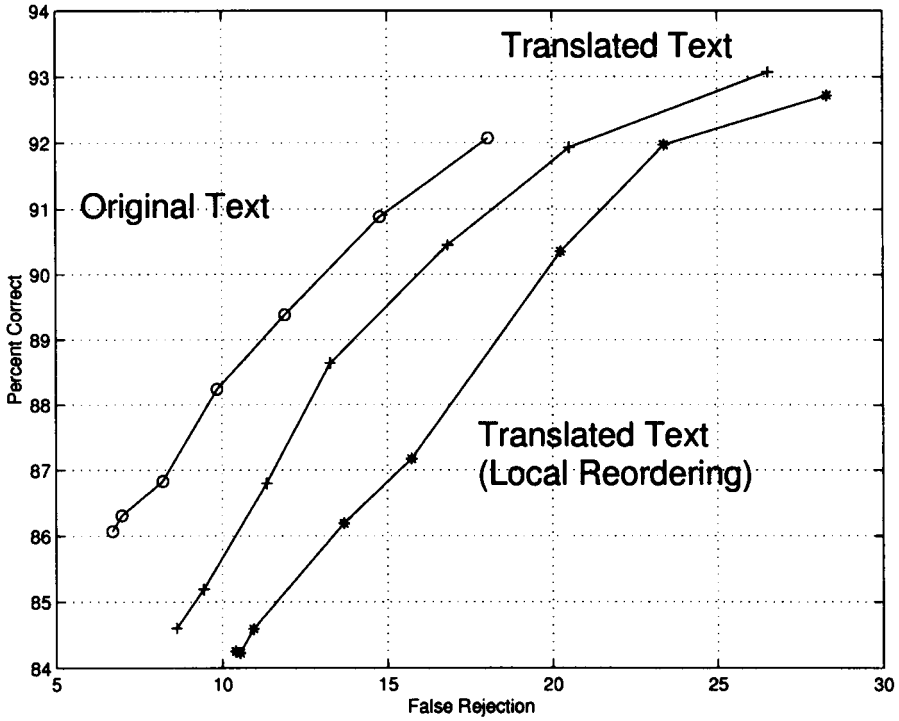

Figure 15. Call routing accuracy on original English text and English text translated from Spanish with only local reordering and with both local and sentence-level reordering.

There are a few interesting observations to be made concerning Figure 14. 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. 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 selection accuracy in Table III. 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 selection is better than that of the phrase trigram-based model as shown in Table III. This difference narrows at lower false rejection rate.

We have also trained a Spanish–English translation model on the Spanish–English version of the training corpus. We used the translation model to translate the Spanish version of the same test corpus and evaluated the call routing performance on the translated English. Figure 15 illustrates the results of this experiment.

The figure shows the call routing performance curves on the original English text, on the translated English text and on the English text without lexical reordering (with only locally reordered phrases). It is interesting to note that the call routing performance improves due to lexical reordering and at a given false rejection rate the loss in performance due to translation is only about 2%, when compared to the performance on the original text.

7. Conclusion

We have presented a mathematical model for speech translation in limited domains based on SFSTs. 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 with application-dependent and application-independent metrics.

Acknowledgements

We would like to thank Richard Cox and Mazin Rahim for their continued support for the work reported in this paper. We would also like to thank Hiyan Alshawi, Shona Douglas, Allen Gorin and Mehryar Mohri for valuable discussions pertaining to this work.

Notes

[1] Our approach is embodied in a system called Anuvaad (http://www.research.att.com/~srini/Anuvaad.html)

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

[3] We consider source and target word alignment subsequences of 1–1, 2–1, 1–2, 1–0 and 0–1 words.

[4] The Japanese string was translated and segmented so that a token boundary in Japanese corresponds to a token boundary in English as selected by human translators.

[5] An exact permutation finite-state model might also be constructed since the numbers of words in the phrases to be reordered are fairly small.

[6] Note that the movement errors are derived after the strings are compared using insertion, deletion and substitution operations.

[7] The state-of-the-art speech recognition performance on conversation speech such as Switchboard data is about 75% word accuracy.

References

Abella, A. and A. L. Gorin: 1997, 'Generating Semantically Consistent Inputs to a Dialog Manager', in *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*, Rhodes, Greece, pp. 1879–1882.

Abney, S.: 1991, 'Parsing by Chunks', in R. Berwick, S. Abney, and C. Tenny (eds), *Principle-Based Parsing: Computation and Psycholinguistics*, Kluwer Academic Publishers, Dordrecht, pp. 257–278.

ALPAC: 1966, *Languages and Machines: Computers in Translation and Linguistics*, Automatic Language Processing Advisory Committee, Division of Behavioral Sciences, National Research Council, National Academy of Sciences, Washington DC.

Alshawi, H., S. Bangalore, and S. Douglas: 1998a, 'Automatic Acquisition of Hierarchical Transduction Models for Machine Translation', in *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, Quebec, pp. 41–47.

Alshawi, H., S. Bangalore, and S. Douglas: 1998b, 'Learning Phrase-Based Head Transduction Models for Translation of Spoken Utterances', in *5th International Conference on Spoken Language Processing (ICSLP98)*, Sydney, pp. 2767–2770.

Arnold, D., L. Sadler, and R. L. Humphreys: 1993, 'Special Issue on Evaluation of MT Systems', *Machine Translation* **8**(1–2).

Bangalore, S. and A. Joshi: 1999, 'Supertagging: An Approach to Almost Parsing', *Computational Linguistics* **25**, 237–265.

Bangalore, S. and G. Riccardi: 2000, 'Stochastic Finite-State Models for Spoken Language Machine Translation', in *ANLP/NAACL 2000 Workshop Embedded Machine Translation Systems*, Seattle, Washington, pp. 52–59.

Bangalore, S. and G. Riccardi: 2001, 'A Finite-State Approach to Machine Translation', in *2nd Meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, Pennsylvania, pp. 135–142.

Brown, P. F., S. A. Della Pietra, V. J. Della Pietra, and R. Mercer: 1993, 'The Mathematics of Machine Translation: Parameter Estimation', *Computational Linguistics* **16**, 263–312.

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

Kaplan, R. and M. Kay: 1994, 'Regular Models of Phonological Rule Systems', *Computational Linguistics* **20**, 331–378.

Knight, K. and Y. Al-Onaizan: 1998, 'Translation with Finite-State Devices', in D. Farwell, L. Gerber, and E. Hovy (eds), *Machine Translation and the Information Soup: Third Conference of the Association for Machine Translation in the Americas, AMTA '98*, Springer, Berlin, pp. 421–437.

Koskenniemi, K. K.: 1984, 'Two-Level Morphology: A General Computation Model for Word-Form Recognition and Production', Ph.D. thesis, University of Helsinki, Finland.

Lavie, A., L. Levin, M. Woszczyna, D. Gates, M. Gavalda, and A. Waibel: 1999, 'The Janus-III Translation System: Speech-to-Speech Translation in Multiple Domains', in *Proceedings of CSTAR Workshop*, Schwetzingen, Germany.

Nederhof, M.-J.: 2000, 'Practical Experiments with Regular Approximation of Context-Free Languages', *Computational Linguistics* **26**, 17–44.

Papineni, K., S. Roukos, T. Ward, and W. Zhu: 2002, 'BLEU: A Method for Automatic Evaluation of Machine Translation', in *40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA, pp. 313–318.

Pereira, F. C. and M. D. Riley: 1997, 'Speech Recognition by Composition of Weighted Finite Automata', in E. Roche and S. Y. Schabes (eds), *Finite State Language Processing*, MIT Press, Cambridge, MA, pp. 431–453.

Pereira, F. C. and R. Wright: 1997, 'Finite-State Approximation of Phrase-Structure Grammars, in E. Roche and Y. Schabes (eds), *Finite-State Language Processing*, MIT Press, Cambridge, MA, pp. 149–173.

Riccardi, G. and S. Bangalore: 1998, 'Automatic Acquisition of Phrase Grammars for Stochastic Language Modeling', *Proceedings of the Sixth Workshop on Very Large Corpora*, Montreal, Quebec, pp. 188–196.

Riccardi, G., E. Bocchieri, and R. Pieraccini: 1995, 'Non Deterministic Tochastic Language Models for Speech Recognition', in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Detroit, MI, pp. 247–250.

Riccardi, G. and A. Gorin: 2000, 'Stochastic Language Adaptation over Time and State in Natural Spoken Dialogue Systems', *IEEE Transactions on Speech and Audio Processing*, **8**, 3–10.

Riccardi, G., A. L. Gorin, A. Ljolje, and M. Riley: 1997, 'A Spoken Language System for Automated Call Routing', in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '97)*, Munich, pp. 1143–1146.

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

Roche, E.: 1999, 'Finite State Transducers: Parsing Free and Frozen Sentences', in A. Kornai (ed.), *Extened Finite State Models of Language*, Cambridge University Press, Cambridge, pp. 108–120.

Vilar, J. M., V. M. Jiménez, J. C. Amengual, A. Castellanos, D. Llorens, and E. Vidal: 1999, 'Text and Speech Translation by Means of Subsequential Transducers', in A. Kornai (ed.), *Extended Finite State Models of Language*, Cambridge University Press, Cambridge, pp. 121–139.

Wahlster, W. (ed.): 2000, *Verbmobil: Foundations of Speech-to-Speech Translation*, Springer, Berlin.

Woszczyna, M., M. Broadhead, D. Gates, M. Gavaldà, A. Lavie, L. Levin, and A. Waibel: 1998, 'A Modular Approach to Spoken Language Translation for Large Domains', in D. Farwell, L. Gerber, and E. Hovy (eds), *Machine Translation and the Information Soup: Third Conference of the Association for Machine Translation in the Americas, AMTA '98*, Springer, Berlin, pp. 31–40.

Wu, D.: 1997, 'Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora', *Computational Linguistics* **23**, 377–404.

Yamamoto, M. and K. W. Church: 1998, 'Using Suffix Arrays to Compute Term Frequency and Document Frequency for All Substrings in a Corpus', in *Proceedings of the Sixth Workshop on Very Large Corpora*, Montreal, Quebec, pp. 28–37.