Finite-State Transducer-based Statistical Machine Translation using Joint Probabilities

Srinivas Bangalore, Stephan Kanthak and Patrick Haffner

AT&T Labs-Research
180 Park Ave
Florham Park, NJ 07932
{srini,kanthak,haffner}@att.research.com

Abstract

In this paper, we present our system for statistical machine translation that is based on weighted finite-state transducers. We describe the construction of the transducer, the estimation of the weights, acquisition of phrases (locally ordered tokens) and the mechanism we use for global reordering. We also present a novel approach to machine translation that uses a maximum entropy model for parameter estimation and contrast its performance to the finite-state translation model on the IWSLT Chinese-English data sets.

1. Introduction

The problem of machine translation 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. In previous work, we have proposed stochastic finite-state transducer (SFST) models for these two subproblems [1, 2] which can then be composed into a single SFST model for Statistical Machine Translation (SMT). SFST approach to SMT has gained momentum in recent years with several groups following this approach successfully [3, 4, 5, 6] for speech to speech translation. The attractions of this approach 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.

In this paper, we present the SFST model for SMT and detail the different phases in training the model and decoding using the model. We also present a novel approach to translation using a discriminatively trained model for lexical choice and a permutation automaton for lexical reordering. This model does not rely on word-level alignments at all which makes it an attractive approach for translating between language pairs that have significantly different word orders (English-Japanese, for example) where the string-based word-alignment methods perform poorly.

The outline of the paper is as follows. In Section 2, we present an overview of the system. In Section 3, we describe in detail the different stages used to train an SFST translation model. We illustrate the steps in the decoding of a source input using the SFST translation in Section 4. In Section 5, we present a discriminative model for translation which addresses the shortcomings of SFST translation.

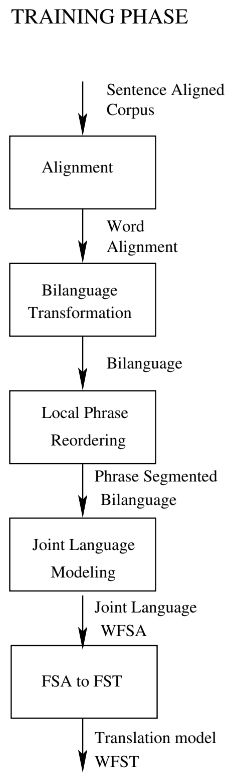

Figure 1: Training phases for our system

2. System Overview

In this section, we present an overview of our system. The details of each component of the system are explained in subsequent sections of the paper.

The training of the SFST model starts with a sentence

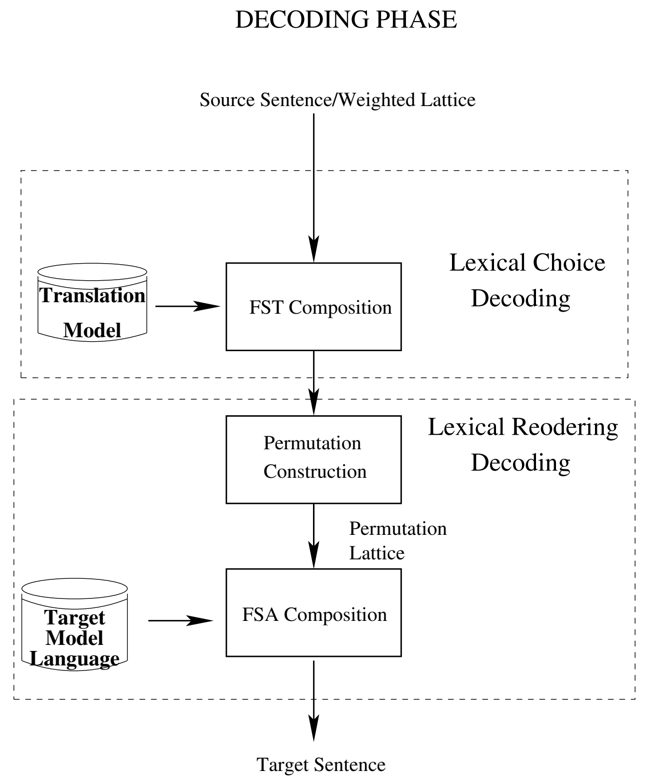

Figure 2: Decoding phases for our system

aligned corpus as the input as shown in Figure 1. The sentence aligned corpus is used to infer a word alignment using an alignment training process. Word alignment results in a mapping between the words in the source language sentence and the words in the target language sentence. Words from either the source or target sentence may be left unmapped. Each word aligned sentence is transformed into a bilanguage string where the first component is a source word and the second component is a target word. The choice of representing the bilanguage in the order of the source language implies that the target language words would not be in the target language order. In order to alleviate this problem, we use local phrase reordering with a fixed window which retokenizes the bilanguage string so as to align the words appearing in the window to be in the correct source and target language order. This retokenized string is used to build a n-gram language model. The n-gram language model approximates the joint probability model between the source sentence and source-ordered target language sentence. The n-gram language model is represented as a weighted finite-state acceptor (FSA) and the arcs are interpreted as having two components, thus resulting a weighted finite-state transducer.

In the decoding/translation phase (Figure 2), a given input, a sentence or a weighted lattice from a speech recognizer is represented as a weighted FSA (WFSA). The WFSA is composed with the SFST from the training process and the highest probability path is extracted as the output of lexical selection phase. The resulting target language sentence is expected to have segments of words that would be in the target language word order, however the phrases may have to be reordered globally to form a well formed target language sentence. We use lexical reordering to transform the output of the lexical selection phase to create a well-formed target language sentence.

3. SFST Training

In this section, we describe each of the components of our lexical selection system in detail.

3.1. Word Alignment

The first stage in the process of training a lexical selection model is obtaining an alignment function that given a pair of source ($s_1 s_2 \dots s_n$) and target ($t_1 t_2 \dots t_m$) language sentences, maps source language word subsequences into target language word subsequences, as shown below.

$$\forall i \exists j (f(s_i) = t_j \vee f(s_i) = \epsilon) \qquad (1)$$

For this purpose, in the past, we have used the tree alignment algorithm described in [7]. For the work reported in this paper, we have used the GIZA++ tool [8] which implements a string-alignment algorithm.

GIZA++ alignment however is asymmetric in that the word mappings are different depending on the direction of alignment – source-to-target or target-to-source. Hence in addition to the functions f as shown in Equation 1 we train another alignment function g as shown in Equation 2.

$$\forall j \exists i (g(t_j) = s_i \vee g(t_j) = \epsilon) \qquad (2)$$

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

Figure 3: Example bilingual texts with alignment information

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

Figure 4: Bilanguage strings resulting from alignments shown in Figure 1.

3.2. Bilanguage Representation

From the alignment information in Figure 3, we construct a bilanguage representation of each sentence in the bilingual corpus. The bilanguage string consists of source-target symbol pair sequences as shown in Equation 3. 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.

$$\begin{aligned} B^f &= b_1^f \, b_2^f \, \ldots \, b_m^f \\ b_i^f &= (s_{i-1}; s_i, f(s_i)) \;\; if \;\; f(s_{i-1}) = \epsilon \\ &= (s_i, f(s_{i-1}); f(s_i)) \;\; if \;\; s_{i-1} = \epsilon \\ &= (s_i, f(s_i)) \;\; otherwise \end{aligned} \tag{3}$$

Figure 4 shows an example alignment and the source-word-ordered bilanguage strings corresponding to the alignment shown in Figure 3.

We also construct a bilanguage using the alignment function g similar to the bilanguage using the alignment function f as shown in Equation 3.

Thus, the bilanguage corpus obtained by combining the two alignment functions is $B = B_f \cup B_g$.

3.3. Bilingual Phrase Acquisition and Local Reordering

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. Also, the use of phrases allows for reordering of words in the phrase to be in correct target language order, thus solving part of the lexical reordering problem. Moreover, SFSTs can take advantage of the phrasal correlation to improve the computation of the probability $P(W_S, W_T)$ [1].

The bilanguage representation could result in multiple words of the source sentence to be mapped to multiple words of the target sentence as a consequence of some words being aligned to ϵ. In addition to these phrases, we compute subsequences of a given length k on the bilanguage string and for each subsequence we reorder the target words of the subsequence to be in the same order as they are in the target language sentence corresponding to that bilanguage string. This results in a retokenization of the bilanguage where some tokens are source-target word pairs and others are source-target phrase pairs.

3.4. SFST Representation

From the bilanguage corpus B, we train a n-gram language model using language modeling tools [9, 10]. The resulting language model is represented as a weighted finite-state automaton ($S \times T \rightarrow [0, 1]$). The symbols on the arcs of this automaton ($s_i\_t_i$) are interpreted as having the source and target symbols (s_i:t_i), making it into a weighted finite-state transducer ($S \rightarrow T \times [0, 1]$) that provides a weighted string-to-string transduction from S into T (as shown in Equation 4).

$$T^* = \underset{T}{argmax} P(s_i, t_i | s_{i-1}, t_{i-1} \ldots s_{i-n-1}, t_{i-n-1}) \tag{4}$$

4. Decoding

In this section, we describe the decoding process and the global reordering process in detail. Since we represent the translation model as a weighted finite state transducer, the decoding process of translating a new source input (sentence or weighted lattice) amounts to a transducer composition and selection of the best probability path resulting from the composition.

$$T^* = \pi_1(BestPath(I_s \circ TransFST)) \tag{5}$$

However, we have noticed that on the development corpus, the resulting target sentence is typically shorter than the intended target sentence. This mismatch may be due to the incorrect estimation of the back-off events and their probabilities in the training phase of the transducer. In order to alleviate this mismatch, we introduce a negative word insertion penalty model as a mechanism to produce more words in the target sentence.

4.1. Word Insertion Model

The word insertion model is also encoded as a weighted finite-state automaton and is included in the decoding sequence as shown in Equation 6. The word insertion FST has one state and $| \sum_T |$ number of arcs each weighted with a λ weight representing the word insertion cost. The word insertion model penalizes or rewards paths which have more words depending on whether λ is positive or negative value.

$$T^* = \pi_1(BestPath(I_s \circ TransFST \circ WIP)) \tag{6}$$

4.2. Global Reordering

Local reordering as described in Section 3.3 is restricted by the window size and accounts only for different word order within phrases. During decoding we apply global reordering by permuting the words of the best translation and weighting the result by an n-gram language model (see also Figure 2):

$$T^* = BestPath(perm(T') \circ LM_t) \tag{7}$$

Unfortunately, even the size of the minimal permutation automaton grows exponentially with the length of the input sequence. While decoding by composition simply resembles the principle of memoization (i.e. in this case: all state hypotheses necessary to decode a sentence are kept in memory), it is necessary to either use heuristic forward pruning or limit the window of the allowed permutations. Similar to the way described in [11] the latter was used here.

Decoding ASR output in combination with global reordering uses either n-best lists or extracts n-best lists from lattices first. Decoding using global reordering is performed for each entry of the n-best list separately and the best decoded target sentence is picked from the union of the n intermediate results.

5. Discriminant Models for Translation

The approach presented in the previous section is a generative model for statistical machine translation. However, discriminant modeling techniques have become the dominant method for resolving ambiguity in speech and natural language processing tasks outperforming generative models for the same task. Discriminative training has been used only for translation model combination [8] but not directly to train the parameters of a model. In recent work [12], we have started to work on discriminatively trained models for lexical selection, which we detail in this section. We expect these models to outperform generative models as well as provide a framework for incorporating rich morpho-syntactic information which result in sparseness for generative models.

Statistical machine translation can be formulated as a search for the best target sequence that maximizes $P(T|S)$, where S is the source sentence and T is the target sentence. Ideally, $P(T|S)$ should be estimated directly to maximize the conditional likelihood on the training data (discriminant model). However, T corresponds to a sequence with a exponentially large combination of possible labels, and traditional classification approaches cannot be used directly. Although, Conditional Random Fields (CRF) [13] train an exponential model at the sequence level, in translation tasks such as ours the computational requirements of training such models is prohibitively expensive.

5.1. Maximum Entropy-based Sequential Lexical Choice Model

We approximate the string level global classification problem, using independence assumptions, to a product of local classification problems as shown in Equations 8.

$$P(T|S) = \prod_{i}^{N} P(t_i|\Phi(S,i)) \tag{8}$$

where $\Phi(S,i)$ is a set of features extracted from the source string S (shortened as Φ in the rest of the section).

A very general technique to obtain the conditional distribution $P(t_i|\Phi(S,i))$ is to choose the distribution with least assumptions (with Maxent) that properly estimates the average of each feature over the training data [14]. This gives us the Gibbs distribution parameterized with the weights λ_t where t ranges over the label set and V is the total number of target language vocabulary.

$$P(t_i|\Phi) = \frac{e^{\lambda_{t_i} \cdot \Phi}}{\sum_{t=1}^{V} e^{\lambda_t \cdot \Phi}} \tag{9}$$

The weights are chosen so as to maximize the conditional likelihood $L = \sum_i L(s_i, t_i)$ with

$$L(S,T) = \sum_i \log P(t_i|\Phi) = \sum_i \log \frac{e^{\lambda_{t_i} \cdot \Phi}}{\sum_{t=1}^{V} e^{\lambda_t \cdot \Phi}} \tag{10}$$

The procedures used to find the global maximum of this concave function include two major families of methods: Iterative Scaling (IS) and gradient descent procedures, in particular L-BFGS methods [15], which have been reported to be the fastest. We obtained faster convergence with a new Sequential L1-Regularized Maxent algorithm (SL1-Max) [16], compared to L-BFGS[1]. We have adapted SL1-Max to conditional distributions for our purposes. Another advantage of the SL1-Max algorithm is that it provides L1-regularization as well as efficient heuristics to estimate the regularization meta-parameters. The computational requirements are $O(V)$ and as all the classes need to be trained simultaneously, memory requirements are also $O(V)$. Given that the actual number of non-zero weights is much lower than the total number of features, we use a sparse feature representation which results in a feasible runtime system.

5.1.1. Frame level discriminant model: Binary Maxent

For the machine translation tasks, even allocating $O(V \cdot F)$ memory (where F denotes the number of features) during training exceeds the memory capacity of current computers. To make learning more manageable we factorize the word-level multi-class classification problem into binary classification sub-problems. This also allows for parallelization during training the parameters. We use here V one-vs-other binary classifiers at each word. Each output label t is projected into a bit string, with components $b_j(t)$. The probability of each component is estimated independently:

$$P(b_j(t)|\Phi) = 1 - P(\bar{b}_j(t)|\Phi) = \frac{1}{1 + e^{-(\lambda_j - \lambda_{\bar{j}}) \cdot \Phi}} \tag{11}$$

where $\lambda_{\bar{j}}$ is the parameter vector for $\bar{b}_j(y)$. Assuming the bit vector components to be independent, we have $P(t_i|\Phi) = \prod_j P(b_j(t_i)|\Phi)$. Therefore, we can decouple the likelihood and train the classifiers independently. In this paper, we use the simplest and most commonly studied code, consisting of V one-vs-others binary components. The independence assumption states that the output labels or classes are independent.

5.2. Maximum Entropy-based Bag-of-Words Lexical Choice Model

The sequential lexical choice model described in the previous section treats the selection of a lexical choice for a source word in the local lexical context as a classification task. The data for training such models is derived from the word alignment corpus obtained from alignment algorithms such as GIZA++. The decoded target lexical items have to be further reordered, but for closely related languages the reordering could be incorporated into correctly ordered target phrases as discussed previously.

For pairs of languages with radically different word order (e.g. English-Japanese), there needs to be a global reorder-

[1]http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

	Training		Dev 2005		Dev 2006	
	Chinese	English	Chinese	English	Chinese	English
Sentences	46,311		506		489	
Running Words	351,060	376,615	3,826	3,897	5,214	6,362*
Vocabulary	11,178	11,232	931	898	1,136	1,134*
Singletons	4,348	4,866	600	538	619	574*
OOVs [%]	-	-	0.6	0.3	0.9	1.0
ASR WER [%]	-	-	-	-	25.2	-

Table 1: Statistics about the training and development data 05 and 06. * = first of multiple reference translations only.

	Dev 2005	Dev 2006	
	Text	Text	ASR 1-best
FST	51.8	19.5	16.5
SeqMaxEnt	53.5	19.4	16.3
BOWMaxEnt	59.9	19.3	16.6

Table 2: Results (mBLEU) scores for the three different models on the transcriptions for development set 2005 and 2006 and ASR 1-best for development set 2006.

ing of words similar to the case in the SFST-based translation system. Also, for such differing language pairs, the alignment algorithms such as GIZA++ perform poorly.

These observations prompted us to formulate the lexical choice problem *without* the need for word alignment information. We require a sentence aligned corpus as before, but we treat the target sentence as a bag-of-words or BOW assigned to the source sentence. The goal is, given a source sentence, to estimate the probability that we find a given word in the target sentence. This is why, instead of producing a target sentence, what we initially obtain is a target bag of words. Each word in the target vocabulary is detected independently, so we have here a very simple use of binary static classifiers. Training sentence pairs are considered as positive examples when the word appears in the target, and negative otherwise. Thus, the number of training examples equals the number of sentence pairs, in contrast to the sequential lexical choice model which has one training example for each token in the bilingual training corpus. The classifier is trained with n-gram features ($BOgrams(S)$) from the source sentence. During decoding the words with conditional probability greater than a threshold θ are considered as the result of lexical choice decoding.

$$BOW_T^* = \{t|P(t|BOgrams(S) > \theta\} \qquad (12)$$

In oder to reconstruct the proper order of words in the target sentence we consider all permutations of words in BOW_T^* and weight them by a target language model. This step is similar to the one described in Section 4.2 and we indeed use the same implementation here.

The bag-of-words approach can also be modified to allow for length adjustments of target sentences, if we add optional deletions in the final step of permutation decoding. The parameter θ and an additional word deletion penalty can then be used to adjust the length of translated outputs.

6. Data and Experiments

We have performed experiments on the IWSLT06 Chinese-English training and development sets from 2005 and 2006. The data are traveler task expressions such as seeking directions, expressions in restaurants and travel reservations. Table 1 presents some statistics on the data sets. It must be noted that while the 2005 development set matches closely with the training data, the 2006 development set has been collected separately and shows slightly different statistics for average sentence length, vocabulary size and out-of-vocabulary words. Also the 2006 development set contains no punctuation marks in Chinese, but the corresponding English translations have punctuation marks. We also evaluated our models on the speech recognition output and we report results on the 1-best output of the speech recognizer. The 1-best Chinese speech recognition word error rate is 25.2%.

For the experiments, we tokenized the Chinese sentence into character strings and trained the models discussed in the previous sections. Also, we trained a punctuation prediction model using maxent framework on the Chinese character strings in order to insert punctuation marks into the 2006 development data set. The resulting character string with punctuation marks is used as input to the translation decoder. For the 2005 development set, punctuation insertion was not needed since the Chinese sentences already had the true punctuation marks.

In Table 2 we present the results of the three different translation models – FST, Sequential Maxent and Bag-of-words Maxent – on the data described above. There are a few interesting observations that can be made based on these results. First, on the 2005 development set, the sequential maxent model outperforms the FST model, even though the two models were trained starting from the same GIZA++ alignment. The difference, however, is due to the fact that maxent models can cope with increased lexical context[2] and the parameters of the model are discriminatively trained. The more surprising result is that the bag-of-words maxent model significantly outperforms the sequence maxent model. The reason is that the sequence maxent model relies on the word alignment, which if erroneous, results in incorrect prediction by the sequential maxent model. The bag-of-words model, on the other hand does not rely on the word-level alignment and can be interpreted as a discriminatively trained model of dictionary lookup for a target word in the context of a source sentence.

The second set of observations relate to the difference in performance between 2005 development set and 2006 development set. As indicated in the data release document, the 2006 set was collected in a very different manner compared to the 2005 set. As a consequence, the mismatch between the training set and 2006 development set in terms of lexical and syntactic difference can be seen precipitating the lower performance. Due to this mismatch, the performance of the maxent models are not very different from the FST model; indicating the lack of good generalization across different genres. We however believe that the maxent framework allows for incorporation of linguistic features that could potentially help in generalization across genres. For translation of ASR 1-best, we see a systematic degradation of about 3% in mBLEU score compared to translating the transcription.

The bag-of-words approach is very promising because it performs reasonably well despite considerable and easy to identify losses of information between the source and the target. The first and most obvious loss is about word position. The only information we use right now to restore the target word position is the global language model, the information about the position in the source sentence is completely lost. We are currently working toward incorporating syntactic information on the target words so as to be able to recover some of the position information lost in the classification process.

A less obvious loss is the number of times a word or concept appears in the target sentence. *Function words* like ”the” and ”of” can appear many times in an English sentence. We are currently exploring several solutions to handle these multiple function words. *Synonyms* are target words which translate the same source word. Suppose that in the training data, target words t_1 and t_2 are, with equal probability, translations of the same source word. Then, in the presence of this source word, the probability to detect the corresponding target word, which is normally 0.8 (we assume some noise), will be, because of discriminant learning, split equally between t_1 and t_2, that is 0.4 and 0.4. Because of this synonym problem, we immediately see that the threshold has to be set lower than 0.5, which is observed experimentally. However, if we set the threshold to 0.3, both t_1 and t_2 will be detected in the target sentence, and we found this to be a major source of undesirable insertions.

[2]We use 6 words to the left and right of a source word for sequential maxent, but only 2 preceding source and target words for FST approach.

7. Conclusions

We view machine translation as consisting of lexical selection and lexical reordering steps. These two steps need not necessarily be sequential and could be tightly integrated. We have presented the weighted finite-state transducer model of machine translation where lexical choice and a limited amount of lexical reordering are tightly integrated into a single transduction. We have also presented a novel approach to translation where these two steps are loosely coupled and the parameters of the lexical choice model are discriminatively trained using a maximum entropy model. The lexical reordering model in this approach is achieved using a permutation automaton. We have evaluated these two approaches on the 2005 and 2006 IWSLT development sets.

8. References

[1] S. Bangalore and G. Riccardi, “Stochastic finite-state models for spoken language machine translation,” in *Proceedings of the Workshop on Embedded Machine Translation Systems*, 2000, pp. 52–59.

[2] S. Bangalore and G. Riccardi, “A Finite-State Approach to Machine Translation,” in *NAACL, Pittsburgh*, 2001.

[3] S. Kanthak and H. Ney, “FSA: An Efficient and Flexible C++ Toolkit for Finite State Automata Using On-Demand Computation,” in *ACL 2004*, Barcelona, Spain, 2004, pp. 510–517.

[4] B. Zhou, S. Chen, and Y. Gao, “Constrained phrase-based translation using weighted finite-state tranducer,” in *ICASSP 2005*, Philadelphia, 2005.

[5] S. Kumar, Y. Deng, and W. Byrne, “A weighted finite state transducer translation template model for statistical machine translation,” *Natural Language Engineering*, vol. 12, no. 1, pp. 35–75, 2006.

[6] J. Vilar, V.M. Jiménez, J. Amengual, A. Castellanos, D. Llorens, and E. Vidal, “Text and speech translation by means of subsequential transducers,” in *Extened Finite State Models of Language*, András Kornai, Ed. Cambridge University Press, 1999.

[7] H. Alshawi, S. Bangalore, and S. Douglas, “Learning Phrase-based Head Transduction Models for Translation of Spoken Utterances,” in *The fifth Interna-*

tional Conference on Spoken Language Processing (ICSLP98), Sydney, 1998.

[8] F. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proceedings of ACL*, 2002.

[9] V. Goffin, C. Allauzen, E. Bocchieri, D. Hakkani-Tur, A. Ljolje, S. Parthasarathy, M. Rahim, G. Riccardi, and M. Saraclar, "The AT&T WATSON Speech Recognizer," in *Proceedings of ICASSP*, Philadelphia, PA, 2005.

[10] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit," in *Proc. Intl. Conf. Spoken Language Processing*, 2002.

[11] S. Kanthak, D. Vilar, E. Matusov, R. Zens, and H. Ney, "Novel reordering approaches in phrase-based statistical machine translation," in *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, Ann Arbor, Michigan, 2005, pp. 167–174.

[12] S. Bangalore, P. Haffner, and S. Kanthak, "Sequence classification for machine translation," in *Interspeech 2006*, Pittsburgh, 2006.

[13] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of ICML*, San Francisco, CA, 2001.

[14] A.L. Berger, Stephen A. D. Pietra, D. Pietra, and J. Vincent, "A Maximum Entropy Approach to Natural Language Processing," *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[15] R. Malouf, "A comparison of algorithms for maximum entropy parameter estimation," in *Proceedings of CoNLL-2002*. 2002, pp. 49–55, Taipei, Taiwan.

[16] M. Dudik, S. Phillips, and R.E. Schapire, "Performance Guarantees for Regularized Maximum Entropy Density Estimation," in *Proceedings of COLT'04*, Banff, Canada, 2004, Springer Verlag.