CONSTRAINED PHRASE-BASED TRANSLATION USING WEIGHTED FINITE-STATE TRANSDUCERS

Bowen Zhou, Stanley F. Chen and Yuqing Gao

IBM T. J. Watson Research Center
Yorktown Heights, New York 10598
{zhou, stanchen, yuqing}@us.ibm.com

ABSTRACT

Phrase-based translation models have shown clear advantages over word-based models, and weighted finite-state transducers (WFST's) provide a unified framework for integrating the various components of a speech-to-speech translation system, such as speech recognition and machine translation. This paper combines these two ideas by proposing a constrained phrase-based statistical machine translation system that we implement using WFST's. We evaluate the proposed model on a bidirectional Chinese-English translation task and show improvements over our previous system.

1. INTRODUCTION

Finite-state methods have been applied in a wide range of speech and language processing applications [1]. Of particular interest are recent efforts in approaching the task of statistical machine translation (SMT) using weighted finite-state transducers (WFST's). Various translation methods have been implemented using WFST's in the literature. For example, Knight et al. [2] describe a system based on word-to-word statistical translation models, Bangalore et al. [3] use WFST's to select and reorder lexical items, and Kumar et al. [4] implement alignment template translation models using WFST's.

One of the reasons why WFST-based approaches are favored is because of the availability of mature and efficient algorithms for general purpose decoding and optimization. For the task of speech-to-speech translation, where our ultimate goal is to obtain a direct translation from speech in a source language to a target language, the WFST framework is even more attractive as it provides the additional advantage of seamlessly integrating speech recognition and machine translation. This framework can be used to incorporate heterogeneous statistical knowledge from multiple sources, by using the composition operation to combine cascaded models expressed as WFST's. This should be particularly valuable when the translation task is complicated by the presence of disfluent conversational speech or recognition errors.

Compared with word-level SMT [5], phrase-based methods explicitly take word context into consideration when translating a word. Koehn et al. [6] compares several schemes as to how to establish phrase-level correspondences, and showed that all of these methods consistently outperform word-based approaches.

The purpose of this paper is to introduce a model for constrained phrase-based SMT that can be rapidly implemented using WFST's. The proposed approach incorporates contextual information explicitly into both the translation and language models. This method differs from existing phrase-based methods primarily in two ways. First, each pair of bilingually-aligned sentences is deterministically partitioned into phrases, so only a single of set of phrases is extracted from each. Thus, only a small number of constrained phrases are generated from a parallel corpus, compared with full phrase-based approaches where a huge set of phrases may be extracted. Second, unlike phrase-based methods where phrase translation probabilities are estimated from relative frequencies derived from word-level alignments, this method first retokenizes the training data using extracted phrases. Next, the retokenized parallel corpus is realigned and the model parameters are estimated based on this new alignment. In addition, the retokenized data is used to train a monolingual language model, so that a phrase-based n-gram language model is obtained.

In this approach, contextual information is captured in two ways. By translating word sequences (or phrases) directly into other word sequences (rather than word-by-word), we preserve the contextual information present *within* phrases. Using a phrase-based language model captures the contextual information present *between* phrases.

The remainder of this paper is organized as follows: Sec. 2 reformulates phrase-based SMT in terms of WFST's. Next, the implementation details of constrained phrase-based SMT using WFST's are discussed in Sec. 3. Experimental results are presented in Sec. 4.

2. A WFST PERSPECTIVE OF SMT

We start by introducing the concept of a *token*. In this paper, a *token* is defined as a semantic unit in the parallel sentence, which can either be a *word* or a *phrase* (i.e., a sequence of words). Specifically for Chinese, a *token* refers to a single segment in a segmented Chinese character sequence.

From the perspective of WFST's, translating a foreign token sequence f_1^J with length J to an English token string e_1^I of length I can be viewed as seeking a stochastic process that maximizes the joint probability $\Pr(e_1^I, f_1^J)$:

$$\hat{e} = \arg\max_{e_1^I} \Pr(e_1^I, f_1^J) \tag{1}$$

This joint probability can be expressed as a sum over all hidden variable values, which can be approximated by maximization:

$$\begin{aligned} \hat{e} &= \arg\max_{e_1^I} \sum_{g_1^I, n_1^I, m_1^I} \Pr(e_1^I, g_1^I, n_1^I, m_1^I, f_1^J) \\ &\approx \arg\max_{e_1^I} \max_{g_1^I, n_1^I, m_1^I} \Pr(e_1^I, g_1^I, n_1^I, m_1^I, f_1^J) \end{aligned} \tag{2}$$

Thanks to the DARPA CAST/Babylon program for funding this work.

where m_1^I is the random variable describing the permutation model, n_1^I is the fertility model which describes how many source words or phrases should be generated for each target token, and g_1^I is the "*NULL* insertion" model which describes for each target token whether a NULL token that corresponds to one or more superfluous tokens in the source language should be inserted before it.

The key observation in implementing SMT using WFST's is that the translation model can be decomposed as a chain of conditional probabilities as follows:

$$
\begin{aligned}
& Pr(e_1^I, g_1^I, n_1^I, m_1^I, f_1^J) = \\
& Pr(e_1^I) \cdot && (3) \\
& Pr(m_1^I | e_1^I) \cdot && (4) \\
& Pr(n_1^I | m_1^I, e_1^I) \cdot && (5) \\
& Pr(g_1^I | n_1^I, m_1^I, e_1^I) \cdot && (6) \\
& Pr(f_1^J | g_1^I, n_1^I, m_1^I, e_1^I) && (7)
\end{aligned}
$$

The conditional probability distributions in Eqs. 3–7 can be estimated from parallel training data using some assumptions and approximations, as described by Brown et al. [5]. Furthermore, each distribution can be represented by a WFST that models the relationship between its input and output, as described in Sec. 3.2. Therefore, the right side of Eq. 2 can be implemented as a cascade of finite-state machines that are connected by the *composition* operation; and the maximum operator in Eq. 2 can be realized through standard Viterbi search on the resulting translation graph.

3. CONSTRAINED PHRASE-BASED SMT VIA WFST

The implementation of constrained phrase-based SMT using WFST's consists of three steps. First, we need to generate aligned phrase pairs from a parallel corpus, and estimate the corresponding translation model and language model from the training data. Second, such models need to be compiled into finite-state machines. Finally, an efficient search is required for fast translation.

3.1. Word Alignment and Phrase Extraction

For Chinese-English translation, the Chinese sentences in the parallel corpus need to be segmented by inserting whitespace between appropriate characters. Chinese text segmentation has been widely studied in various language processing tasks, and it should be noted that what defines the "best" segmentation of a given sentence may depend on the target application. In our task, our goal is to extract monolingual phrases rather than individual words; i.e., we consider each segmented token to be a Chinese phrase. In our proposed method, phrases are first defined for Chinese and then English phrases are learned from the parallel corpus. To this end, we prefer longer Chinese segments subject to the constraint that the corresponding English words form a consecutive sequence.

The segmentation is performed using a stack decoder that maximizes the probability of the sequence of segmented tokens. The sequence probability is modeled by a trigram language model with a vocabulary of 32,000 Chinese phrases. In addition, we add in a length penalty factor that encourages the search to prefer longer phrases.

Next, we collect English phrases from the parallel corpus using the segmentation we have for the Chinese data. To this end, word-level alignment is first carried out using the GIZA++ toolkit [7] that implements the "IBM models" proposed by Brown et al. [5]. As noted by Och and Ney [7], the baseline IBM models possess the limitation that they do not allow a source token to be aligned with two or more target tokens. Therefore, we first align the training data in the reverse direction, i.e., from English to Chinese, so that each Chinese token is allowed to map to multiple English tokens. Since longer phrases are preferred in our Chinese segmentation, it is typical that more than one English word is aligned with each Chinese token. English phrases are then extracted based on the Viterbi alignment of the parallel corpus: whenever a Chinese token is aligned with a sequence of adjacent English words, this sequence is selected as a candidate English phrase. There is no limitation on the length of candidate phrases.

These grouped English word sequences are collected along with their absolute frequencies in the aligned parallel corpus. A high frequency indicates a strong tendency for the word sequence to occur as a phrase in the *given* parallel corpus, and we thus retain such word sequences as phrases. For low-frequency word sequences, we compare their frequency in aligned text with their frequency in text ignoring alignments; those word sequences with high relative frequency are also preserved. Finally, the remaining low-frequency word sequences are intersected with a domain-specific phrase dictionary and the overlapping entries are retained.

Next, the training corpus is retokenized using the induced English phrases. For example, the text "*chronic medical concerns*" is tokenized as the single token "*chronic_medical_concerns*" in the training data. Note that we replace all occurrences of a word sequence with its corresponding phrase, whether or not the word sequence aligns to a single Chinese token in the given sentence pair. In this way, retokenization may correct previous alignment errors.

The retokenized parallel corpus is then aligned again in both directions, i.e., Chinese to English and English to Chinese. The intersection of these two alignments is computed to improve alignment accuracy [7], followed by a number of heuristics that extend alignments by including likely correspondences between English and Chinese tokens. In addition to the expansion heuristics discussed by Koehn et al. [6], we also include the alignment between $\bar{e}_i$ and $\bar{f}_j$ if the current estimate of $P(\bar{f}_j|\bar{e}_i)$ is larger than some threshold. Given this refined alignment, the fertility and translation models are reestimated by collecting the relevant relative frequencies. The same retokenization technique is also applied to monolingual language model training data, to build appropriate constrained phrase-based language models.

Compared with methods where translation probabilities for collected phrase pairs are estimated from relative frequencies computed from word-level alignments [6], we believe that phrase-based alignments may significantly modify alignment probabilities, including fertility and translation probabilities.

3.2. WFST Cascades for SMT

Corresponding to Eqs. 3–7, the following five WFST's need to be constructed to perform a translation task: The acceptor L assigns probabilities to target language strings based on a back-off trigram language model, the transducer N models the fertilities of tokens in its input; the transducer G describes the probability of generating source tokens from the *NULL* token; the transducer T determines the probabilities of mapping target tokens to source ones; and the transducer M encodes the allowable reorderings of source tokens (so as to be in the same order as the target tokens they align to).

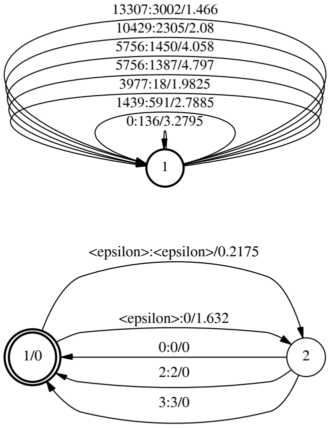

Fig. 1. A portion of the translation transducer T (top) and *NULL* insertion transducer G (bottom). Tokens are encoded using integer indices.

Then, the translation of an input S can be computed by finding the best path in the following lattice:

$$D = S \circ M \circ ((N \circ G \circ T)^{-1} \circ L) \qquad (8)$$

where the WFST $H = (N \circ G \circ T)^{-1} \circ L$ is independent of the input and can be constructed off-line so as to improve decoding speed. Here, "$\circ$" is the composition operator and "-1" represents the inversion operation.

We note that WFST's representing SMT models are generally not determinizable, and the presence of a large number of transitions with ϵ labels in the component WFST's may cause the computation of H to be difficult given typical memory constraints. This is particularly an issue for applications with large vocabularies and language models, and special consideration is required in constructing transducers for such tasks.

We implemented the transducers T and G similarly as in [2]; portions of these WFST's are depicted in Fig. 1. Both the input and output labels of G indicate numeric target token ID's. The input and output labels of T represent target and source token ID's, respectively.

The natural fertility model for target token sequences inherently introduces a closure that will produce infinity ambiguity. In our system, the transducer N is implemented as shown in Fig. 2 in order to avoid ϵ loops, which can cause memory issues during composition. Each branch out of the start state (at the far left) corresponds to a different token and its fertility probabilities; for a given path from start to final state, the number of times an input token is replicated in the output encodes its fertility. Thus, tokens 2 and 4 have fertilities of 0, 1, or 2; and token 3 has fertilities of 0, 1, 2, 3, or 4. In practice, the fertility transducer is pruned by only considering fertilities with probability $n(\phi|\bar{e}_i) > 0.01$, to reduce memory demands. Fig. 2 shows the fertility WFST for a three-token system after pruning.

The cost associated with a transition is taken to be the negative logarithm of the corresponding probability. We use the Viterbi paradigm (a.k.a. the tropical semiring); i.e., when two paths with the same labels are merged, the resulting cost is the minimum of the individual path costs. Minimization is performed following each composition in computing H. These operations were all carried out using the IBM finite-state machine toolkit [8].

3.3. On-the-Fly Decoding via WFST's

In our stand-alone on-line translation decoder, the input source sentence is first reordered by a permutation transducer M, which permits each token to be distorted only within a window of size 3 centered at its original position. In addition, swapping the first and last token of a sentence is also allowed, to account for word-order differences between Chinese and English for questions.

To improve the speed of on-line composition and the subsequent search for the best path, we perform *lazy* composition followed by pruning with a threshold α, so that only promising states in D are expanded. Next, Viterbi search is applied on the pruned graph to find the lowest cost path. Details of decoder parameter setting and speed are provided in Sec. 4.2.

4. EXPERIMENTAL EVALUATION

We evaluated the proposed translation system on a two-way Chinese-English speech translation task in the domain of medical care. The objective of this system is to facilitate conversation between an English-speaking doctor and Chinese-speaking patients.

4.1. Corpus and Setup

The training corpus was collected by IBM and other participants of the DARPA Babylon/CAST program. The majority of the corpus was collected from simulated English-English doctor/patient interactions, and the dialogues was later translated into Chinese. There are about 128K utterance pairs in this corpus (C1). We note that the Chinese translations in C1 may not be representative of conversational Chinese. Therefore, around 6,000 spoken sentences (C2) were collected directly from a native Chinese community, to better capture the linguistic characteristics of conversational Chinese. After being transcribed and translated into English, this set of data was also included in our corpus.

Several randomly selected dialogues from C1 and C2 were reserved for evaluation; we refer to this data as test set 1 (T1). To better simulate a realistic testing environment, the Chinese-to-English (C2E) test set was selected entirely from C2 (582 sentences with 4 reference translations each), which is more conversational and thus more challenging. Similarly, the doctor's side of several dialogues selected from C1 (300 sentences with 2 reference translations each) was reserved to evaluate English-to-Chinese (E2C) translation. The rest of the data was used as the training corpus. No punctuation marks are present in either the training or testing data. Further evaluation was performed using the canned data provided by the CAST program. The text-based data set is manually transcribed from English-Chinese conversations, along with 8 human translations that are used as reference translations. We refer to this data as test set 2 (T2). Table 1 summarizes some statistics of the training and test corpora.

The training process described in Sec 3.1 generated a vocabulary of 10,401 unique Chinese tokens and 14,069 unique English tokens, of which 4,621 are phrases ranging in length from two to six words. The sizes of the WFST's built from the trained models are listed in Table 2.

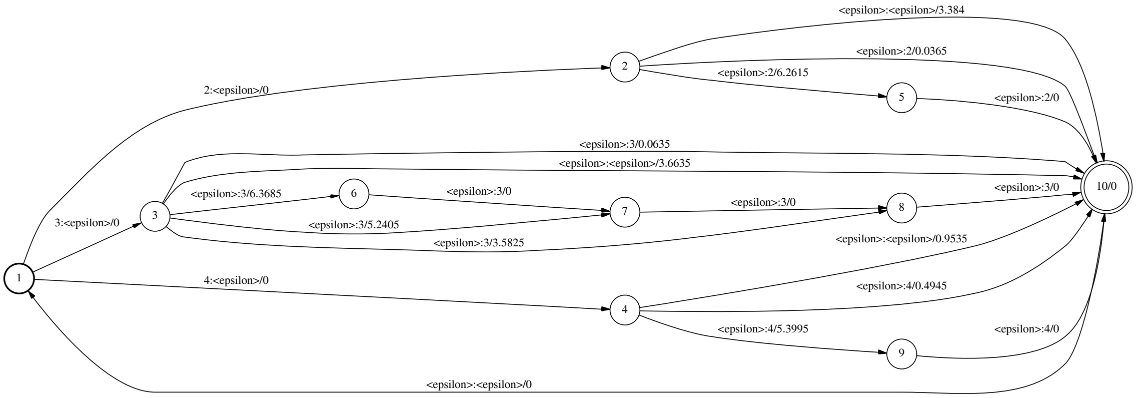

Fig. 2. The fertility WFST N for a 3-token system.

Table 1. Corpora statistics.

Data	English	Chinese
Training set	134K sentences	
	5.3 word/sent.	7.4 character/sent.
Test set 1 (T1)	300 sentences	582 sentences
	7.1 word/sent.	8.9 character/sent.
Test set 2 (T2)	132 sentences	73 sentences
	6.1 word/sent.	6.2 character/sent.

Table 2. The size of WFST's for various models.

WFST	# of states		# of transitions	
	c2e	e2c	c2e	e2c
T	1	1	67,980	52,689
G	2	2	14,073	10,405
N	60,695	36,176	122,684	75,139
L	23,822	23,182	150,581	146,165
H	207,552	164,435	10,375,799	9,970,106

4.2. Experimental Results

We first adjusted the pruning threshold α applied during lazy composition in decoding development data, to find a good balance between translation speed and performance. For the results in Table 3, the average decoding speed on a 2.4 GHz Pentium 4 CPU was less than 2 seconds per sentence for all tasks.

Experimental results are presented in Table 3 in terms of the BLEU metric [9]. The translation results are compared with the output of the NLU+NLG (natural language understanding and generation) system that we proposed in an earlier study [10]. We note that for the NLU+NLG method, only 16K annotated sentence pairs (A1, which is a subset of C1) were used to train the statistical parsing and generation models, as these models can only be trained on parsed data. We observe from Table 3 that the proposed WFST approach achieves superior performance over these two test sets for both translation directions. For T1, the WFST approach obtains a larger gain; one reason may be that T1 is better represented in the complete training set (C1+C2) than in A1 due to its larger size. This effect may be even stronger for the English portion of T1, and thus the proposed WFST approach achieves a larger improvement in E2C translation.

Table 3. Evaluation of translation performance: BLEU score.

	Test Set T1	Test Set T2
C2E WFST	0.2597	0.2862
C2E NLU+NLG	0.2221	0.2785
E2C WFST	0.3266	0.2244
E2C NLU+NLG	0.2536	0.2152

5. REFERENCES

[1] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech and Language*, vol. 16, no. 1, pp. 69–88, 2002.

[2] K. Knight and Y. Al-Onaizan, "Translation with finite-state devices," in *4th AMTA Conference*, 1998.

[3] S. Bangalore and G. Ricardi, "A finite-state approach to machine translation," in *NAACL*, 2001.

[4] S. Kumar and W. Byrne, "A weighted finite state transducer implementation of the alignment template model for statistical machine translation," in *HLT/NAACL*, 2003.

[5] P. Brown, V. Della Pietra, S. Della Pietra, and R. Mercer, "The mathematics of statistical machine translation: parameter estimation," *Comput. Linguist.*, vol. 19, no. 2, pp. 263–311, 1993.

[6] P. Koehn, F. Och, and D. Marcu, "Statistical phrase-based translation," in *NAACL/HLT*, 2003.

[7] F. J. Och and H. Ney, "Improved statistical alignment models," in *ACL00*, Hong Kong, China, October 2000, pp. 440–447.

[8] S. Chen, "The IBM finite-state machine toolkit," Technical Report, IBM T. J. Watson Research Center, 2000.

[9] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "Bleu: a method for automatic evaluation of machine translation," Technical Report RC22176, IBM T. J. Watson Research Center, 2001.

[10] B. Zhou, Y. Gao, J. Sorensen, Z. Diao, and M. Picheny, "Statistical natural language generation for trainable speech-to-speech machine translation systems," in *ICSLP02*, Denver, CO, Sept. 2002.