LATTICE-BASED VITERBI DECODING TECHNIQUES FOR SPEECH TRANSLATION

George Saon and Michael Picheny

IBM T. J. Watson Research Center, Yorktown Heights, NY, 10598
e-mail: gsaon@us.ibm.com

ABSTRACT

We describe a cardinal-synchronous Viterbi decoder for statistical phrase-based machine translation which can operate on general ASR lattices (as opposed to confusion networks). The decoder implements constrained source reordering on the input lattice and makes use of an outbound distortion model to score the possible reorderings. The phrase table, representing the decoding search space, is encoded as a weighted finite state acceptor which is determinized and minimized. At a high level, the search proceeds by performing simultaneous transitions in two pairs of automata: (input lattice, phrase table FSM) and (phrase table FSM, target language model). An alternative decoding strategy that we explore is to break the search into two independent subproblems: first, we perform *monotone* lattice decoding and find the best *foreign* path through the ASR lattice and then, we decode this path *with reordering* using standard sentence-based SMT.

We report experimental results on several testsets of a large scale Arabic-to-English speech translation task in the context of the Global Autonomous Language Exploitation (or GALE) DARPA project. The results indicate that, for monotone search, lattice-based decoding outperforms 1-best decoding whereas for search with reordering, only the second decoding strategy was found to be superior to 1-best decoding. In both cases, the improvements hold only for shallow lattices.

1. INTRODUCTION

Current research in speech translation focuses on augmenting the interface between ASR and MT to more than 1-best hypotheses, the idea being that the MT component should have the freedom to select paths whose translations are more likely. We distinguish between the *cascaded approach* to speech translation, where the system is comprised of a series of individual engines that are applied in sequence, and the *integrated approach*, where the SMT component accepts multiple ASR hypotheses as input. In increasing order of generality, these hypotheses can be presented in various forms: n-best lists, confusion networks [1] and arbitrary lattices [2, 3]. The maximum degree of generality (or the tightest coupling) is attained by performing a joint ASR and MT search which has only been possible, so far, for limited domain tasks [4].

There is no guarantee, at the outset, that lattice-based SMT is bound to improve translation performance. This is because ASR lattices contain paths which are worse than the 1-best in terms of word error rate in addition to the paths which are better. This is an often overlooked aspect of lattice processing and researchers tend to focus only on the oracle word error rate (i.e. the error rate of the path of lowest WER). It can happen, however, that a weak MT component prefers paths which result in worse overall performance. Therefore, care has to be taken on how to combine the ASR and MT scores when doing lattice decoding.

While lattice-based SMT decoding has been studied previously already [2, 3, 4], the work presented here exhibits differences along several axes. First, compared to the prior art, we discuss how to deal with word reordering on a lattice during decoding. Secondly, the way we construct the search space (subsection 3.3) also differentiates our work. Thirdly, to the best of our knowledge, we are the first to apply lattice decoding to a truly large scale speech translation task.

The remainder of this paper is organized as follows: section 2 formulates the problem, section 3 describes the lattice-based SMT decoder, section 4 presents some experimental results on a large scale Arabic-to-English speech translation task and section 5 summarizes our findings.

2. PROBLEM FORMULATION

We are given $\mathbf{x} = x_1 \dots x_T$, $x_t \in \mathbb{R}^d$, a sequence of acoustic feature vectors corresponding to a foreign speech utterance which is to be translated into a target string $\mathbf{e} = e_1 \dots e_I$, $e_i \in E$. The goal of speech translation is to find the target string $\hat{\mathbf{e}}$ with the highest posterior probability given the acoustics $\mathbf{x}$:

$$\hat{\mathbf{e}} = \underset{\mathbf{e} \in E^*}{\operatorname{argmax}}\, P(\mathbf{e}|\mathbf{x}) \tag{1}$$

The difference with text translation is that the foreign sequence is not given, instead we have to sum up over all possible realizations $\mathbf{f} = f_1 \dots f_J$, $f_j \in F$ as suggested in [5]:

$$P(\mathbf{e}|\mathbf{x}) = \sum_{\mathbf{f} \in F^*} P(\mathbf{e}, \mathbf{f}|\mathbf{x}) \tag{2}$$

By applying Bayes rule to the term on the right side of equation (2), we get:

$$\begin{aligned} P(\mathbf{e}, \mathbf{f}|\mathbf{x}) &= \frac{P(\mathbf{x}|\mathbf{e}, \mathbf{f}) P(\mathbf{e}, \mathbf{f})}{P(\mathbf{x})} \\ &= \frac{P(\mathbf{x}|\mathbf{f}) P(\mathbf{e}, \mathbf{f})}{P(\mathbf{x})} \end{aligned} \tag{3}$$

where in the last equation of (3) we made the (reasonable) assumption that the acoustic sequence does not depend on the target word sequence when conditioned on the foreign word sequence.

By combining (1), (2) and (3) and by taking into consideration that $P(\mathbf{x})$ is constant with respect to the $\max$ operation, we obtain:

We would like to acknowledge the support of DARPA under Grant HR0011-06-2-0001 for funding part of this work.

$$\begin{aligned}\hat{\mathbf{e}} &= \operatorname*{argmax}_{\mathbf{e}\in E^*} \sum_{\mathbf{f}\in F^*} P(\mathbf{x}|\mathbf{f})P(\mathbf{e},\mathbf{f}) \\ &= \operatorname*{argmax}_{\mathbf{e}\in E^*} \sum_{\mathbf{f}\in F^*} P(\mathbf{x}|\mathbf{f})P(\mathbf{f}|\mathbf{e})P(\mathbf{e})\end{aligned} \tag{4}$$

with $P(\mathbf{x}|\mathbf{f})$ being the likelihood of the acoustic feature vectors given the foreign word sequence, $P(\mathbf{f}|\mathbf{e})$ the likelihood of the foreign word sequence given the target words and $P(\mathbf{e})$, the prior probability of the target word sequence (given by the English language model). In (4) we have applied the source-channel model for machine translation [6] to separate the two knowledge sources: the LM term which controls the well-formedness of the target word sequences and the translation model. Alternatively, it is also common, especially when modeling the translation problem with WFSTs, to reason directly in terms of joint probabilities as in [7, 3].

2.1. Lattice-based decoding

Lattice-based decoding simply means that we restrict the summation in (4) only to the paths occurring in a lattice, that is:

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}\in E^*} \sum_{\mathbf{f}\in L(\mathbf{x})} P(\mathbf{x}|\mathbf{f})P(\mathbf{f}|\mathbf{e})P(\mathbf{e}) \tag{5}$$

where $L(\mathbf{x})$ denotes the set of paths that occur in the ASR lattice corresponding to the acoustics $\mathbf{x}$.

2.2. Phrase-based SMT

In phrase-based translation [8], we segment the foreign input sentence into a sequence of $1 \leq K \leq J$ phrases $\bar{\mathbf{f}} = \bar{f}_1 \dots \bar{f}_K$. We assume a uniform distribution over all possible segmentations. Each foreign phrase $\bar{f}_k$ in $\bar{\mathbf{f}}$ is translated into a target phrase $\bar{e}_k$ with probability $\phi(\bar{f}_k|\bar{e}_k)$. The conditional probability $P(\mathbf{f}|\mathbf{e})$ can be expressed as:

$$\begin{aligned}P(\mathbf{f}|\mathbf{e}) &= \sum_{\bar{\mathbf{f}},\bar{\mathbf{e}}} P(\bar{\mathbf{f}}|\bar{\mathbf{e}}) \\ &= \sum_{\bar{\mathbf{f}},\bar{\mathbf{e}}} \prod_{k=1}^{K} \phi(\bar{f}_k|\bar{e}_k)\end{aligned} \tag{6}$$

where the summation is performed over all the segmentations which are consistent with $\mathbf{f}$ and $\mathbf{e}$.

2.3. Source word reordering

Let $\sigma : \{1,\dots,J\} \rightarrow \{1,\dots,J\}$, $\sigma \in S_J$, be a permutation of the set $\{1,\dots,J\}$ with S_J being the symmetric group of order J. We define the permuted foreign sequence as

$$\mathbf{f}_\sigma := f_{\sigma(1)} \dots f_{\sigma(J)}$$

The translation model can be expressed by marginalizing over the permutations and by using the chain rule of conditional probabilities as follows:

$$\begin{aligned}P(\mathbf{e},\mathbf{f}) &= \sum_{\sigma\in S_J} P(\mathbf{e},\mathbf{f},\sigma) \\ &= \sum_{\sigma\in S_J} P(\mathbf{f}|\mathbf{e},\sigma)P(\mathbf{e}|\sigma)P(\sigma) \\ &= \sum_{\sigma\in S_J} P(\mathbf{f}_\sigma|\mathbf{e})P(\mathbf{e})P(\sigma)\end{aligned} \tag{7}$$

with $P(\sigma)$ being given by the distortion model. The distortion model used in this work is based on the window/skip model introduced in [9] and the scores are computed following the *outbound* distortion scheme proposed in [10] which, for the sake of completeness, is written below:

$$P(\sigma) := \prod_{j=1}^{J-1} P(\sigma(j+1) - \sigma(j)|f_{\sigma(j)}) \tag{8}$$

which simply means that we model the probability of the length of the jump to the next word conditioned on the identity of the current word. Note that this particular distortion scheme depends on both σ and $\mathbf{f}$.

Putting (5) and (7) together, we can formulate the lattice-based SMT decoding problem with reordering in the following way:

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}\in E^*} \sum_{\mathbf{f}\in L(\mathbf{x})} P(\mathbf{x}|\mathbf{f}) \sum_{\sigma\in S_J} P(\mathbf{f}_\sigma|\mathbf{e})P(\mathbf{e})P(\sigma) \tag{9}$$

2.4. Two-pass strategy

Implementing (9) directly can be computationally expensive because of the double summation. The idea then is to divide the problem into two subparts in the following manner:

1. Lattice-based *monotone* decoding of best *foreign* path:

$$\hat{\mathbf{f}} = \operatorname*{argmax}_{\mathbf{f}\in L(\mathbf{x})} \max_{\mathbf{e}\in E^*} P(\mathbf{x}|\mathbf{f})P(\mathbf{f}|\mathbf{e})P(\mathbf{e}) \tag{10}$$

2. Sentence-based *non-monotone* decoding of best *English* path:

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}\in E^*} \sum_{\sigma\in S_J} P(\hat{\mathbf{f}}_\sigma|\mathbf{e})P(\mathbf{e})P(\sigma) \tag{11}$$

This strategy has the obvious advantage of factoring the search into subproblems of lower complexity. In the next section, we discuss how to implement equations (9)–(11) efficiently with WFSTs.

3. DECODER DESCRIPTION

3.1. WFST interpretation

Finite state machines offer a convenient formalism to represent, access and manipulate heterogeneous knowledge sources in a uniform way. Knowledge sources can be combined through the composition operation and the resulting automata can be "shrunk" to an optimal size via determinization and minimization and efficiently searched. It is no wonder then that WFSTs have lately become the formalism of choice in both ASR [11] and SMT [3, 2, 7] and we shall make no exception here.

Similarly to [7], equation (9) can be rewritten using finite-state terminology as follows:

$$\hat{\mathbf{e}} = \text{best-path}(I \circ R \circ M \circ L) \tag{12}$$

where the component FSMs represent:

- I: input lattice acceptor. Encodes $\mathbf{f}$ with weight $P(\mathbf{x}|\mathbf{f})P(\mathbf{f})$.
- R: reordering transducer. Maps $\mathbf{f}$ to $\mathbf{f}_\sigma$ with weight $P(\sigma)$.
- M: translation transducer. Maps $\mathbf{f}_\sigma$ to $\mathbf{e}$ with weight $P(\mathbf{f}_\sigma|\mathbf{e})$.
- L: language model acceptor. Encodes $\mathbf{e}$ with weight $P(\mathbf{e})$.

3.2. Decoding approaches

Computing the entire composition in (12) off-line is usually intractable for a larger scale task. The alternative is to have a decoder which implements *on-demand* (or on-the-fly) composition during the search. This approach has been favored by [7, 3]. Both authors opt for using general FSM decoders. This has the advantage that the same decoder can be used for different FSM configurations and the disadvantage that the decoder is oblivious to special FSMs (e.g. R). We take a different approach and design a specialized decoder which has an efficient implementation for the reordering FSM (but cannot support arbitrary FSM configurations).

3.3. Search space construction

The starting point for creating the M FSM is a bilingual phrase table with scores. Next, we carry out the following steps in sequence which are illustrated in Figure 1 for a French-to-English example:

1. Add one path per phrase from the start to the final state
2. Determinize resulting acceptor
3. Minimize previous acceptor
4. Make FSM cyclic and mark foreign/English arcs

We encoded the phrase table as an acceptor (instead of a transducer) because the determinization and minimization operations are much simpler on acyclic acceptors and we rely on the decoder to differentiate between the foreign and the English arcs.

la maison bleue	the blue house	0.3
la maison rouge	the red house	0.2
une maison bleue	a blue house	0.2
une maison rouge	a red house	0.3

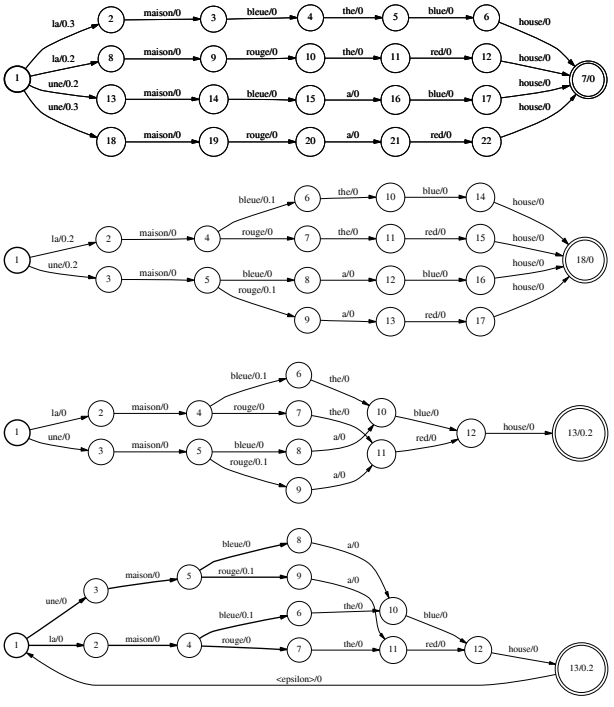

Figure 1: Steps for constructing a phrase-table acceptor.

3.4. Viterbi decoding

The search proceeds in the following way. At each step, we take simultaneous transitions in pairs of FSMs which are labeled by the same word: first, we take one transition in (I, M) and then, if possible, we take a sequence of transitions in (M, L). Note that this requires the "English" section of the phrase-table FSM to be sorted in topological order. The search stops when we reach the final states in all component FSMs. All hypotheses which compete at a given step cover the same number of input words which is commonly referred to as *cardinal synchronous* decoding. These hypotheses are pruned using a combination of beam and histogram (or rank) pruning as explained in [11] for ASR.

3.5. Monotone decoding

For monotone decoding, we keep track of tuples of active states (l_i, m_j, n_k), where l_i is a state in the input lattice, m_j is a state in the phrase table FSM and n_k represents a target language model state which is similar to [3]. In addition, we keep track of the following component scores which are combined with independent weights: acoustic and foreign language model scores from the ASR, translation scores and English LM scores.

3.6. Non-monotone decoding

In the case of search with reordering, there are a couple of additional complications. First, we have to be able to access distant successors in a lattice from any given node, (e.g. immediate successors have distance 1, their immediate successors have distance 2, and so on). Second, we have to keep track of more information during the search for each hypothesis:

- Current state in input lattice
- First state in input lattice from which reordering window applies
- State vector for reordering window
- Jump distance from previous word (used for reordering)
- State in translation FSM
- State in LM FSM

The state vector for the reordering window is an extension of the coverage bit vector [7] except that, instead of bits, we have lattice states. It can be represented as a 64 bit integer if we limit ourselves to a maximum window size of 4 and if each lattice has less than 64K nodes. This elegantly solves the problem of path recombination i.e. hypotheses which have identical window values can be merged.

Compared to monotone decoding, there is an additional distortion score which is also combined with an independent weight.

4. EXPERIMENTS AND RESULTS

The experiments were conducted on a large scale Arabic-to-English broadcast news and broadcast conversations speech translation task which is part of the GALE DARPA program. We report results on four test sets which vary by collection time and by genre. Table 1 summarizes the characteristics of these various test sets.

The speech recognition system used in this work consists in the first two decoding passes of our 2006 GALE evaluation system (which had three passes/models). The first pass search is done with speaker independent acoustic models and the second pass

Name	Genre	Nb. segments	Nb. words
BNAD05s	BN	8	15.0K
DEV07	BN+BC	110	19.6K
EVAL06s-BN	BN	24	6.0K
EVAL06s-BC	BC	14	6.7K

Table 1: Testset description. Nb. of segments refers to the number of shows or "snippets". Nb. of words represents the number of *decoded* Arabic words.

BNAD05s	DEV07	EVAL06s-BN	EVAL06s-BC
15.7%	17.9%	24.0%	31.9%

Table 2: ASR 1-best word error rates.

with speaker-adaptive trained (or SAT) models. Both sets of models are trained discriminatively on 135 hours of supervised data and 1800 hours of unsupervised data (i.e. without reference transcripts). The models are unvowelized (or graphemic) in the sense that short Arabic vowels are not explicitly represented. More details about the training of the Arabic models can be found in [12].

We generated two sets of lattices which differ in the average link density (ALD) and the degree of pruning: one set with an ALD of 2.5 (pruned with a beam of 0.5) and another one with an ALD of 5 (pruned with a beam of 1.0). The original (unpruned) lattices have an ALD of 450. The ASR decoder that was used to generate the lattices is described in [11]. The 1-best word error rates on the different test sets are summarized in Table 2.

The MT phrase tables were trained on a variety of corpora: UN parallel corpus, LDC News and various GALE data releases. Phrases were extracted using the *inverse projection constraint* described in [13]. The English language model is a 5-gram LM and has about 800M n-grams. All non-monotone decodings were run with a window width of 4 words and a maximum skip length of 2.

In Table 3 we report TER results for both monotone and non-monotone decodings for 1-best, lattice set 1 (LS1) and lattice set 2 (LS2). We also compare direct decoding with the two-pass strategy presented in subsection 2.4. An analysis of these results show that the largest gains (1 TER point) are obtained for EVAL06s-BC followed by the BNAD05s testset (1 point monotone, 0.5 with reordering). The gains on the other test sets are inconclusive. A second observation is that the two-pass strategy outperforms the direct decoding strategy and degrades less on the larger lattices. Finally, we note that the only testset which improves for the larger lattices is EVAL06s-BC (which also has the highest WER/TER).

5. CONCLUSION

We described a Viterbi decoder for speech translation which operates on general ASR lattices. The decoder finds the path of minimum cost through an on-demand composition of several automata. We deal efficiently with word reordering on the lattice by representing the reordering FSM implicitly as a node coverage vector. A two-pass decoding strategy is also presented which has the advantage of factoring the search into simpler subproblems. Future work will extend some of these ideas to joint decoding.

Monotone decoding				
	BNAD05s	DEV07	EVAL06s-BN	EVAL06s-BC
1bst	64.0%	60.1%	67.6%	70.6%
LS1	63.0%	59.8%	67.4%	69.8%
LS2	63.2%	59.8%	67.8%	69.5%
Non-monotone direct decoding				
1bst	61.7%	58.9%	66.9%	70.3%
LS1	61.5%	59.1%	67.4%	70.8%
LS2	61.6%	59.4%	68.1%	70.4%
Non-monotone two-pass decoding				
LS1	61.2%	58.8%	67.1%	69.9%
LS2	61.3%	59.2%	67.5%	69.3%

Table 3: TER results for lattice-based SMT decoding.

6. ACKNOWLEDGMENT

The authors wish to thank A. Ittycheriah and Y. Al-Onaizan from the Statistical MT group at IBM for help with translation resources.

7. REFERENCES

[1] W. Shen, R. Zens, N. Bertoldi, and M. Federico, "On the integration of speech recognition and statistical machine translation," in *IWSLT-06*, 2006.

[2] L. Mathias and W. Byrne, "Statistical phrase-based speech translation," in *ICASSP-06*, 2006.

[3] B. Zhou, L. Besacier, and Y. Gao, "On efficient coupling of ASR and MT for speech translation," in *ICASSP-07*, 2007.

[4] E. Matusov, S. Kanthak, and H. Ney, "On the integration of speech recognition and statistical machine translation," in *Interspeech-05*, 2005.

[5] H. Ney, "Speech translation: coupling of recognition and translation," in *ICASSP-99*, 1999.

[6] P. Brown, V. Dell Pietra, A. Della Pietra, and R. Mercer, "The mathematics of statistical machine translation: parameter estimation," *Computational Linguistics*, 1993.

[7] S. Kanthak, D. Vilar, E. Matusov, R. Zens, and H. Ney, "Novel reordering approaches in phrase-based statistical machine translation," in *ACL Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, 2005.

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

[9] C. Tillmann and H. Ney, "Word reordering and a dynamic programming beam search algorithm for statistical machine translation," *Computational Linguistics*, 2003.

[10] Y. Al-Onaizan and K. Papineni, "Distortion models for statistical machine translation," in *21st International Conference on Computational Linguistics*, 2006.

[11] G. Saon, D. Povey, and G. Zweig, "Anatomy of an extremely fast LVCSR decoder," in *Interspeech-05*, 2005.

[12] H. Soltau, G. Saon, B. Kingsbury, J. Kuo, L. Mangu, D. Povey, and G. Zweig, "The IBM 2006 GALE Arabic ASR system," in *ICASSP-07*, 2007.

[13] F. Och and H. Ney, "Improved statistical alignment models," in *38th Annual Meeting of the ACL*, 2000.