Efficient Decoding for Statistical Machine Translation with a Fully Expanded WFST Model

Hajime Tsukada
NTT Communication Science Labs.
2-4 Hikaridai Seika-cho Soraku-gun
Kyoto 619-0237
Japan
tsukada@cslab.kecl.ntt.co.jp

Masaaki Nagata
NTT Cyber Space Labs.
1-1 Hikari-no-Oka Yokosuka-shi
Kanagawa 239-0847
Japan
nagata.masaaki@lab.ntt.co.jp

Abstract

This paper proposes a novel method to compile statistical models for machine translation to achieve efficient decoding. In our method, each statistical submodel is represented by a weighted finite-state transducer (WFST), and all of the submodels are expanded into a composition model beforehand. Furthermore, the ambiguity of the composition model is reduced by the statistics of hypotheses while decoding. The experimental results show that the proposed model representation drastically improves the efficiency of decoding compared to the dynamic composition of the submodels, which corresponds to conventional approaches.

1 Introduction

Recently, research on statistical machine translation has grown along with the increase in computational power as well as the amount of bilingual corpora. The basic idea of modeling machine translation was proposed by Brown et al. (1993), who assumed that machine translation can be modeled on noisy channels. The source language is encoded from a target language by a noisy channel, and translation is performed as a decoding process from source language to target language.

Knight (1999) showed that the translation problem defined by Brown et al. (1993) is NP-complete. Therefore, with this model it is almost impossible to search for optimal solutions in the decoding process. Several studies have proposed methods for searching suboptimal solutions. Berger et al. (1996) and Och et al. (2001) proposed such depth-first search methods as stack decoders. Wand and Waibel (1997) and Tillmann and Ney (2003) proposed breadth-first search methods, i.e. beam search. Germann (2001) and Watanabe and Sumita (2003) proposed greedy type decoding methods. In all of these search algorithms, better representation of the statistical model in systems can improve the search efficiency.

For model representation, a search method based on *weighted finite-state transducer* (WFST) (Mohri et al., 2002) has achieved great success in the speech recognition field. The basic idea is that each statistical model is represented by a WFST and they are composed beforehand; the composed model is optimized by WFST operations such as determinization and minimization. This fully expanded model permits efficient searches. Our motivation is to apply this approach to machine translation. However, WFST optimization operations such as determinization are nearly impossible to apply to WFSTs in machine translation because they are much more ambiguous than speech recognition. To reduce the ambiguity, we propose a WFST optimization method that considers the statistics of hypotheses while decoding.

Some approaches have applied WFST to statistical machine translation. Knight and Al-Onaizan (1998) proposed the representation of IBM model 3 with WFSTs; Bangalore and Riccardi (2001) studied WFST models in call-routing tasks, and Kumar and Byrne (2003) modeled phrase-based translation by WFSTs. All of these studies mainly focused on the representation of each submodel used in machine translation. However, few studies have focued on the integration of each WFST submodel to improve the decoding efficiency of machine translation.

To this end, we propose a method that expands all of the submodels into a composition model, reducing the ambiguity of the expanded model by the statistics of hypotheses while decoding. First, we explain the translation model (Brown et al., 1993; Knight and Al-Onaizan, 1998) that we used as a base for our decoding research. Second, our proposed method is introduced. Finally, experimental results show that our proposed method drastically improves decoding efficiency.

2 IBM Model

For our decoding research, we assume the IBM-style modeling for translation proposed in Brown et al. (1993). In this model, translation from Japanese

f to English e attempts to find the e that maximizes $P(e|f)$. Using Bayes' rule, $P(e|f)$ is rewritten as

$$argmax_e P(e|f) \quad = \quad argmax_e P(f|e)P(e),$$

where $P(e)$ is referred to as a language model and $P(f|e)$ is referred to as a translation model. In this paper, we use word trigram for a language model and IBM model 3 for a translation model.

The translation model is represented as follows considering all possible word alignments.

$$P(f|e) \quad = \quad \sum_a P(f, a|e).$$

The IBM model only assumes a one-to-many word alignment, where a Japanese word f in the j-th position connects to the English word e in the a_j-th position.

The IBM model 3 uses the following $P(f, a|e)$.

$$\begin{aligned} P(f, a|e) \quad = \quad & \left(\begin{array}{c} m - \phi_0 \\ \phi_0 \end{array} \right) p_0^{m-2\phi_0} (1 - p_0)^{\phi_0} \cdot \\ & \prod_{i=0}^{l} \phi_i! \, n(\phi_i|e_i) \cdot \\ & \prod_{j=1}^{m} t(f_j|e_{a_j}) d(j|a_j, m, l). \end{aligned} \tag{1}$$

ϕ_i the a number of f words connecting to e_i, and it is called *fertility*. Note, however, that ϕ_0 is the number of words connecting to null words. $n(\phi|e_i)$ is conditional probability where English word e_i connects to ϕ words in f. $n(\phi|e_i)$ is called fertility probability. $t(f_j|e_i)$ is conditional probability where English word e_i is translated to Japanese word f_j and called translation probability. $d(j|i, l, m)$ is conditional probability where the English word in the i-th position connects to the the Japanese word in the j-th position on condition that the length of the English sentence e and Japanese sentence f are l and m, respectively. $d(j|i, l, m)$ is called distortion probability. In our experiment, we used the IBM model 3 while assuming constant distortion probability for simplicity.

3 WFST Cascade Model

WFST is a finite-state device in which output symbols and output weights are defined as well as input symbols. Using *composition* (Pereira and Riley, 1997), we can obtain the combined WFST $T_1 \circ T_2$ by connecting each output of T_1 to an input of T_2. If we assume that each submodel of Equation (1) is represented by a WFST, a conventional decoder can be considered to compose submodels dynamically.

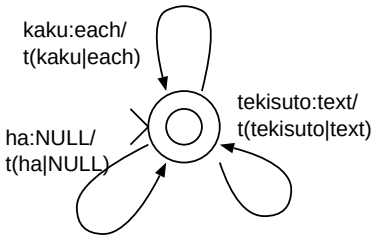

Figure 2: T Model

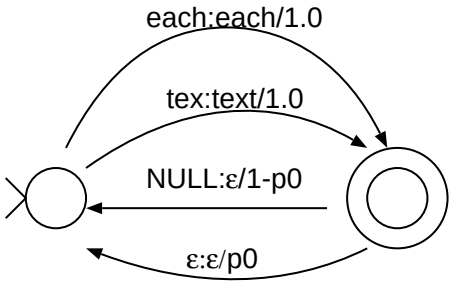

Figure 3: NULL Model

The main idea of the proposed approach is to compute the composition beforehand.

Figure 1 shows the translation process modeled by a WFST cascade. This WFST cascade model (Knight and Al-Onaizan, 1998) represents the IBM model 3 described in the previous section. Any possible permutations of the Japanese sentence are inputed to the cascade. First, *T model*(T) translates the Japanese word to an English word. *NULL model*(N) deletes special word NULL. *Fertility model*(F) merges the same continuous words into one word. At each stage, the probability represented by the weight of a WFST is accumulated. Finally, the weight of *language model* (L) is accumulated. If WFST I represents all permutations of the input sentence, decoding can be considered to search for the best path of $I \circ T \circ N \circ F \circ L$. Therefore, computing $T \circ N \circ F \circ L$ in advance can improve the efficiency of the decoder.

For T, N, and F, we adopt the representation of Knight and Al-Onaizan (1998). For L, we adopt the representation of Mohri et al. (2002). Figures 2–5 show examples of submodel representation with WFSTs. $b(x)$ in Figure 5 stands for a back-off parameter. Conditional branches are represented by nondeterministic paths in the WFST.

4 Ambiguity Reduction

If we can determinize a fully-expanded WFST, we can achieve the best performance of the decoder.

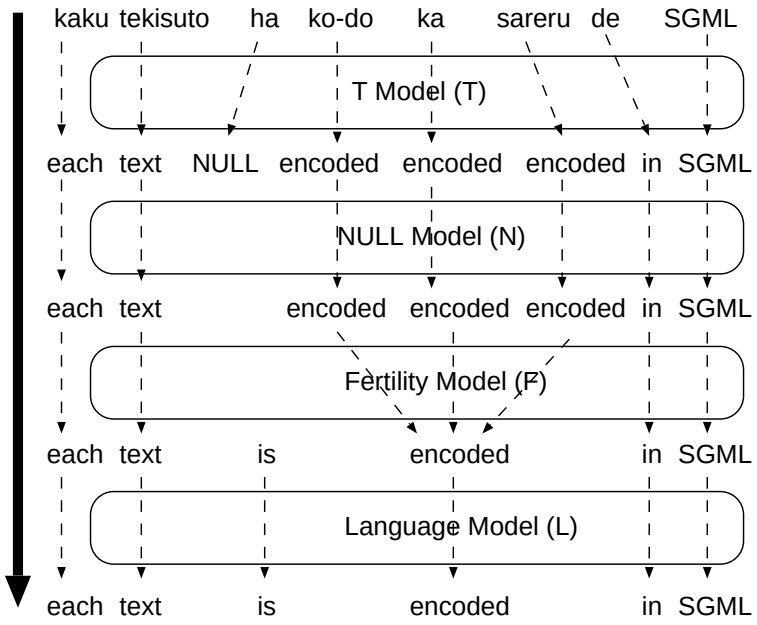

Figure 1: Translation with WFST Cascade Model

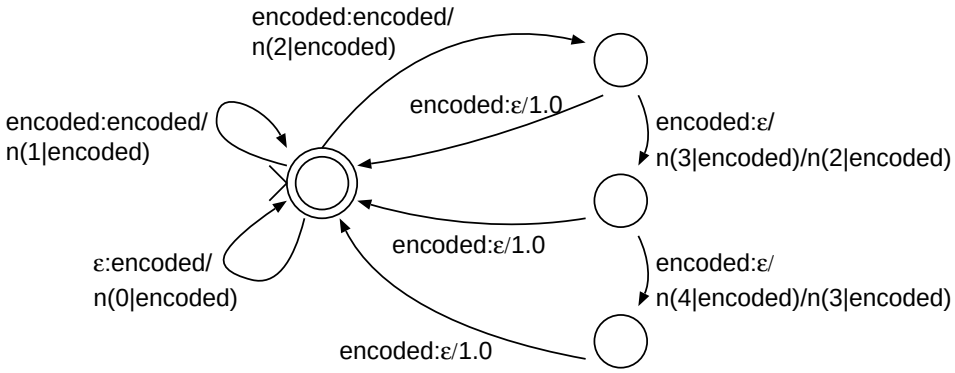

Figure 4: Fertility Model

However, the composed WFST for machine translation is not obviously determinizable. The word-to-word translation model T strongly contributes to WFST's ambiguity while the ϵ transition of other submodels also contributes to ambiguity. Mohri et al. (2002) proposed a technique that added special symbols allowing the WFST to be determinizable. Determinization using this technique, however, is not expected to achieve efficient decoding in machine translation because the WFSTs of machine translation are inherently ambiguous.

To overcome this problem, we propose a novel WFST optimization approach that uses decoding information. First, our method merges WFST states by considering the statistics of hypotheses while decoding. After merging the states, redundant edges whose beginning states, end states, input symbols, and output symbols are the same are also reduced. IBM models consider all possible alignments while a decoder searches for only the most appropriate alignment. Therefore, there are many redundant states in the full-expansion WFST from the viewpoint of decoding.

We adopted a standard decoding algorithm in the speech recognition field, where the forward is beam-search and the backward is A^* search. Since beam-search is adopted in the forward pass, the obtained results are not optimal but suboptimal. All input permutations are represented by a finite-state acceptor (Figure 6), where each state corresponds to input positions that are already read. In the forward search, hypotheses are maintained for each state of

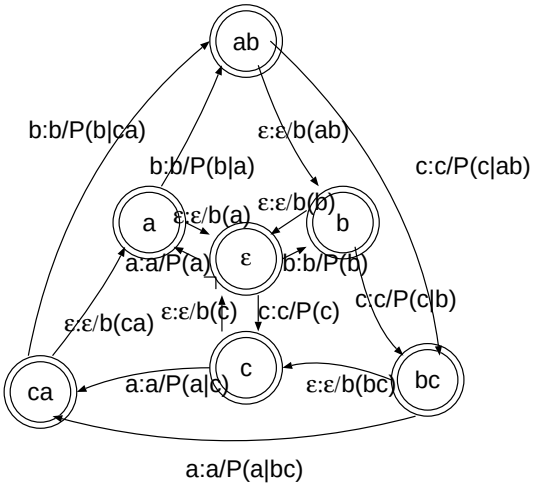

Figure 5: Trigram Language Model

the finite-state acceptor.

The WFST states that always appear together in the same hypothesis list of the forward beam-search should be equated if the states contribute to correct translation. Let M be a full-expansion WFST model and Ref_f be a WFST that represents the correct translation of an input sentence f. For each f, the states of M that always appear together in the same hypothesis list in the course of decoding f with $M \circ Ref_f$ are merged in our method. Simply merging states of M may increase model errors, but Ref_f corrects the errors caused by merging states.

Unlike ordinary FSA minimization, states are merged without considering their successor states. If the weight represents probability, thesum of the weights of output transitions may not be 1.0 after merging states, and then thecondition of probability may be destroyed. Since the decoder does not sum up all possible paths but searches for the most appropriate paths, this kind of state merging does not pose a serious problem in practice.

In the following experiment, we measured the association between states by ϕ^2 in Gale and Church (1991). ϕ^2 is a χ^2-like statistic that is bounded between 0 and 1. If the ϕ^2 of two states is higher than the specified threshold, these two states are merged. The definition of ϕ^2 is as follows, where $a = freq(q_1, q_2)$, $b = freq(q_1) - a$, $c = freq(q_2) - a$, and $d = N - a - b - c$. N is the total number of hypothesis lists. $freq(q)$ ($freq(q_1, q_2)$) is the number of hypothesis lists in which q appears (both q_1 and q_2 appear).

$$\phi^2 = \frac{(ad - bc)^2}{(a + b)(a + c)(b + d)(c + d)}.$$

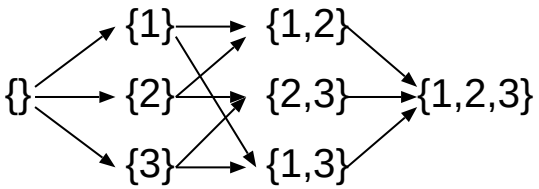

Figure 6: FSA for All Input Permutations

Merging the beginning and end states of a transition whose input is ϵ (ϵ transition for short) may cause a problem when decoding. In our implementation, weight is basically minus log probability, and its lower bound is 0 in theory. However, there exists negative ϵ transition that originated from the back-off value of n-gram. If we merge the beginning and end states of the negative ϵ transition, the search process will not stop due to the negative ϵ loop. To avoid this problem, we rounded the negative weight to 0 if the negative ϵ loop appears during merging.

In the preliminary experiment, a weight-pushing operation (Mohri and Riley, 2001) was also effective for deleting negative ϵ transition of our full-expansion models. However, pushing causes an imbalance of weights among paths if the WFST is not deterministic. As a result of this imbalance, we cannot compare path costs when pruning. In fact, our preliminary experiment showed that pushed full-expansion WFST does not work well. Therefore, we adopted a simpler method to deal with a negative ϵ loop as described above.

5 Experiments

5.1 Effect of Full Expansion

To clarify the effectiveness of a full-expansion approach, we compared the computational costs while using the same decoder with both dynamic composition and static composition, a full-expansion model in other words. In the forward beam-search, any hypothesis whose probability is lower than $1/10$ of the top of the hypothesis list is pruned. In this experiment, permutation is restricted, and words can be moved 6 positions at most. The translation model was trained by *GIZA++* (Och and Ney, 2003), and the trigram was trained by the *CMU-Cambridge Statistical Language Modeling Toolkit v2* (Clarkson and Rosenfeld, 1997).

For the experiment, we used a Japanese-to-English bilingual corpus consisting of example sentences for a rule-based machine translation system. Each language sentence is aligned in the corpus. The total number of sentence pairs is 20,204. We used 17,678 pairs for training and 2,526 pairs

for the test. The average length of Japanese sentences was 8.4 words, and that of English sentences was 6.7 words. The Japanese vocabulary consisted of 15,510 words, and the English vocabulary was 11,806 words. Table 1 shows the size of the WFSTs used in the experiment. In these WFSTs, special symbols that express beginning and end of sentence are added to the WFSTs described in the previous section. The NIST score (Doddington, 2002) and BLEU Score (Papineni et al., 2002) were used to measure translation accuracy.

Table 2 shows the experimental results. The full-expansion model provided translations more than 10 times faster than conventional dynamic composition submodels without degrading accuracy. However, the NIST scores are slightly different. In the course of composition, some paths that do not reach the final states are produced. In the full-expansion model these paths are trimmed. These trimmed paths may cause a slight difference in NIST scores.

5.2 Effect of Ambiguity Reduction

To show the effect of ambiguity reduction, we compared the translation results of three different models. Model O is the full-expansion model described above. Model R is a reduced model by using our proposed method with a 0.9 ϕ^2 threshold. Model $R2$ is a reduced model with the statistics of the decoder without using the correct translation WFST. In other words, $R2$ reduces the states of the full-expansion model more roughly than R. The ϕ^2 threshold for $R2$ is set to 0.85 so that the size of the produced WFST is almost the same as R. Table 3 shows the model size. To obtain decoder statistics for calculating ϕ^2, all of the sentence pairs in the training set were used. When obtaining the statistics, any hypothesis whose probability is lower than $1/10^{0.5}$ of the top of the hypothesis list is pruned in the forward beam-search.

The translation experiment was conducted by successively changing the beam width of the forward search. Figures 7 and 8 show the results of the translation experiments, revealing that our proposed model can reduce the decoding time by approximately half. This model can reduce decoding time to a much greater extent than the rough reduction model, indicating that our state merging criteria are valid.

6 Conclusions

We proposed a method to compile statistical models to achieve efficient decoding in a machine translation system. In our method, each statistical submodel is represented by a WFST, and all submodels are composed beforehand. To reduce the ambiguity of the composed WFST, the states are merged according to the statistics of hypotheses while decoding. As a result, we reduced decoding time to approximately $1/20$ of dynamic composition of submodels, which corresponds to the conventional approach.

In this paper, we applied the state merging method to a fully-expanded WFST and showed the effectiveness of this approach. However, the state merging method itself is general and independent of the fully-expanded WFST. We can apply this method to each submodel of machine translation. More generally, we can apply it to all WFST-like models, including HMMs.

Acknowledgements

We would like to thank F. J. Och for providing *GIZA++* and *mkcls* toolkits, and P. R. Clarkson for *the CMU-Cambridge statistical language modeling toolkit v2*. We also thank T. Hori for providing the n-gram conversion program for WFSTs and F. Bond and S. Fujita for providing the bilingual corpus.

References

Srinivas Bangalore and Giueseppe Riccardi. 2001. A finite-state approach to machine translation. In *Proc. of North American Association of Computational Linguistics (NAACL 2001)*, May.

Adam L. Berger, Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Andrew S. Kehler, and Robert L. Mercer. 1996. Language translation apparatus and method of using context-based translation models. *United States Patent.*

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pitra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

P.R. Clarkson and R. Rosenfeld. 1997. Statistical language modeling using the cmu-cambridge toolkit. In *Proc. of European Conference on Speech Communication and Technology (EUROSPEECH'97)*.

G. Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proc. of HLT 2002.*

William A. Gale and Kenneth W. Church. 1991. Identifying word correspondences in parallel texts. In *Proc. of Fourth DARPA Speech and Natural Language Processing Workshop*, pages 152–157.

Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. 2001. Fast de-

	# of States	# of Transitions
T Model (T)	3	59,026
NULL Model (N)	4	11,810
Fertility Model (F)	91,513	194,360
Language Model (L)	14,532	30,140
Full Expansion($T \circ N \circ F \circ L$)	233,045	2,452,621

Table 1: Submodel/Full-Expansion Model Size

	NIST Score	BLEU Score	Decoding Time (sec.)
Static Composition (Full-Expansion Model)	3.4	0.037	6,596
Dynamic Composition (Conventional Method)	3.5	0.037	84,753

Table 2: Static / Dynamic Composition

coding and optinal decoding for machine translation. In *Proc. of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 228–235, July.

Kevin Knight and Yaser Al-Onaizan. 1998. Translation with finite-state devices. In *Proc. of the 4th AMTA Conference.*

Kevin Knight. 1999. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615.

Shankar Kumar and William Byrne. 2003. A weighted finite state transducer implementation of the alignment template model for statistical machine translation. In *Proc. of Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 142–149, May - June.

Mehryar Mohri and Michael Riley. 2001. A weight pushing algorithm for large vocabulary speech recognition. In *Proc. of European Conference on Speech Communication and Technology (EUROSPEECH'01)*, September.

Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. 2002. Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, 16(1):69–88.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och, Nicola Ueffing, and Hermann Ney. 2001. An efficient A^* search algorithm for statistical machine translation. In *Proc. of the ACL2001 Workshop on Data-Driven Machine Translation*, pages 55–62, July.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLUE: a method for automatic evaluation of machine translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, July.

Fernando Pereira and Michael Riley. 1997. Speech recognition by composition of weighted finite automata. In Emmanuel Roche and Yves Schabes, editors, *Finite-State Language Processing*, chapter 15, pages 431–453. MIT Press, Cambridge, Massachusetts.

Christoph Tillmann and Hermann Ney. 2003. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 29(1):97–133, March.

Ye-Yi Wang and Alex Waibel. 1997. Decoding algorithm in statistical machine translation. In *Proc. of the 35th Annual Meeting of the Association for Computational Linguistics.*

Taro Watanabe and Eiichiro Sumita. 2003. Example-based decoding for statistical machine translation. In *Proc. of MT Summit IX*.

	# of States	# of Transitions
Proposed Model (R)	183,432	2,278,096
Rough Reduction Model ($R2$)	182,212	2,345,255
Original Model (O)	233,045	2,452,621

Table 3: Original/Reduction Model Size

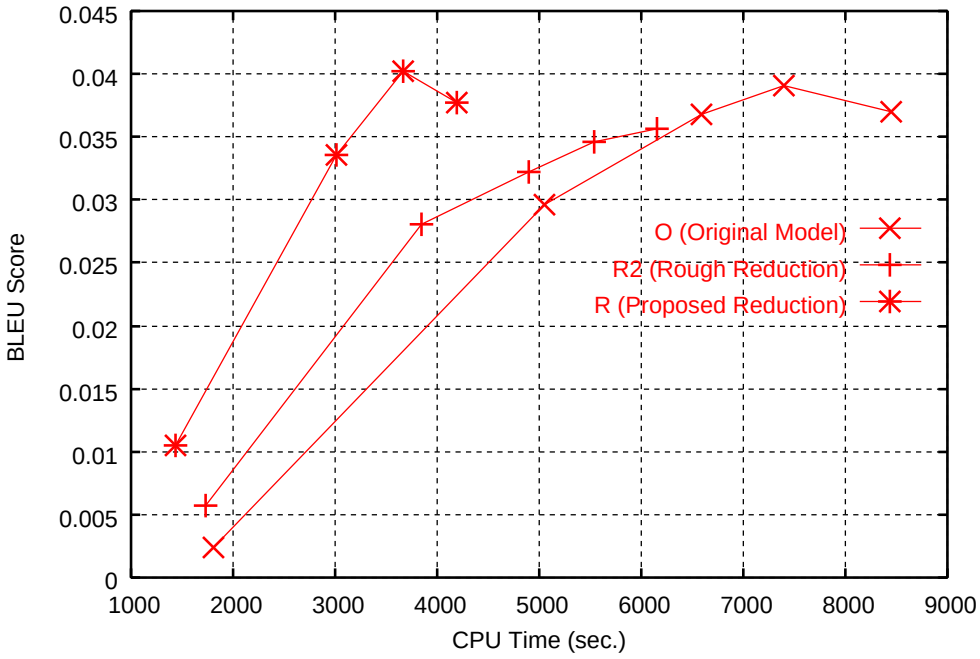

Figure 7: Ambiguity Reduction (BLEU)

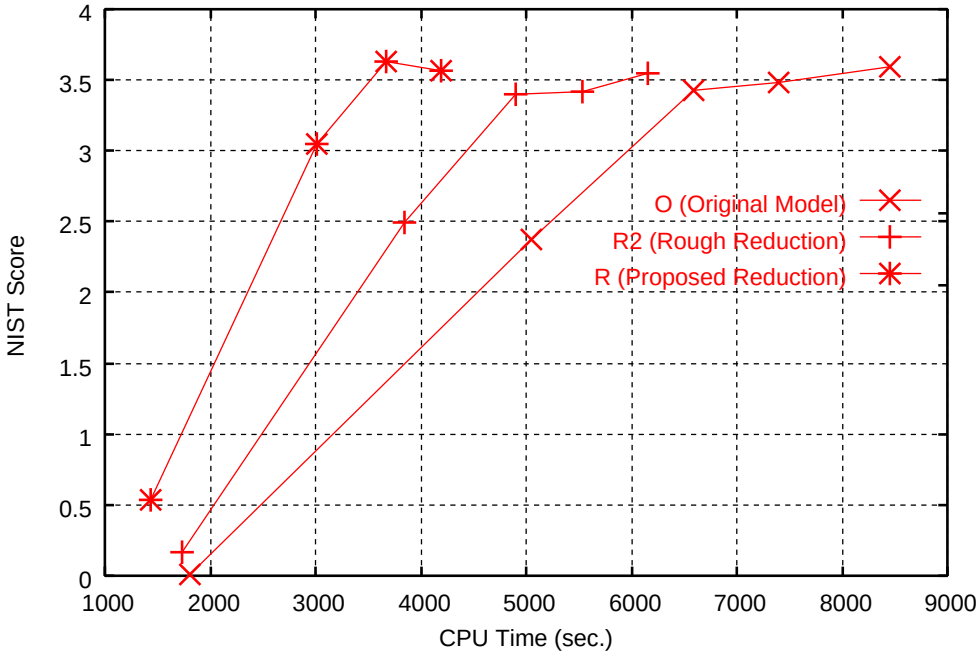

Figure 8: Ambiguity Reduction (NIST)