Phrase-based Machine Translation using Multiple Preordering Candidates

Yusuke Oda† **Taku Kudo**‡ **Tetsuji Nakagawa**‡ **Taro Watanabe**‡
†Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma-shi, Nara 630-0192, Japan
‡Google Japan Inc.
6-11-1 Roppongi, Minato-ku, Tokyo 106-6108, Japan
oda.yusuke.on9@is.naist.jp {taku, tnaka, tarow}@google.com

Abstract

In this paper, we propose a new decoding method for phrase-based statistical machine translation which directly uses multiple preordering candidates as a graph structure. Compared with previous phrase-based decoding methods, our method is based on a simple left-to-right dynamic programming in which no decoding-time reordering is performed. As a result, its runtime is very fast and implementing the algorithm becomes easy. Our system does not depend on specific preordering methods as long as they output multiple preordering candidates, and it is trivial to employ existing preordering methods into our system. In our experiments for translating diverse 11 languages into English, the proposed method outperforms conventional phrase-based decoder in terms of translation qualities under comparable or faster decoding time.

1 Introduction

One of the main problem of phrase-based statistical machine translation (PBMT) (Koehn et al., 2003; Och and Ney, 2004) is handling the difference of word orders between source and target languages. Decoding-time reordering models (Koehn et al., 2005; Zens and Ney, 2006; Galley and Manning, 2008) measure positional relationship between each phrase at the decoding time. However, reordering models have a common problem in that it is difficult to take global information in the source sentence into account, and as a result the decoder may generate grammatically incorrect word orders. In addition, using reordering models demands a complicated decoding algorithm, in which the decoder has to consider concatenations of source phrases with arbitrary orders.

On the other hand, preordering methods (Xia and McCord, 2004; Isozaki et al., 2010; Neubig et al., 2012; Nakagawa, 2015) change word orders of source sentence to be close to the target sentence before starting the decoding process. These methods can use global information in the source sentence and may generate grammatically correct reordering results. However, previous PBMT methods with preordering usually take only one-best preordered sentence and it is difficult to avoid the noise of the input caused by the errors from preordering methods.

One of the trivial way to avoid preordering errors is to obtain N-best preordering candidates, translate each candidate one-by-one and select the most probable result (Li et al., 2007; Zhu, 2014). This method has an obvious problem on computation time because the decoding process is executed N times. Another way to resolve preordering errors is combining a preordering method and decoding-time reordering models. However, it is not trivial to integrating these methods in a single system while comprehending their interactions.

In this study, we propose a new phrase-based decoding method which employs multiple preordering candidates for a source sentence. Our method first encodes multiple preordering candidates as a compact graph structure (we call it *preordering lattice*), and generates translations by a single-pass traversal on the preordering lattice which can take into account all preordering candidates.

Several previous work proposed decoding methods using graph structures with respect to preordering (Niehues and Kolss, 2009; Herrmann et al., 2013a; Herrmann et al., 2013b); however, these methods are tightly integrated with a specific graph structure defined on top of the methods themselves. Another previous work focused on multi-source translation based on a confusion network of multiple source

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 1419–1428, Osaka, Japan, December 11-17 2016.

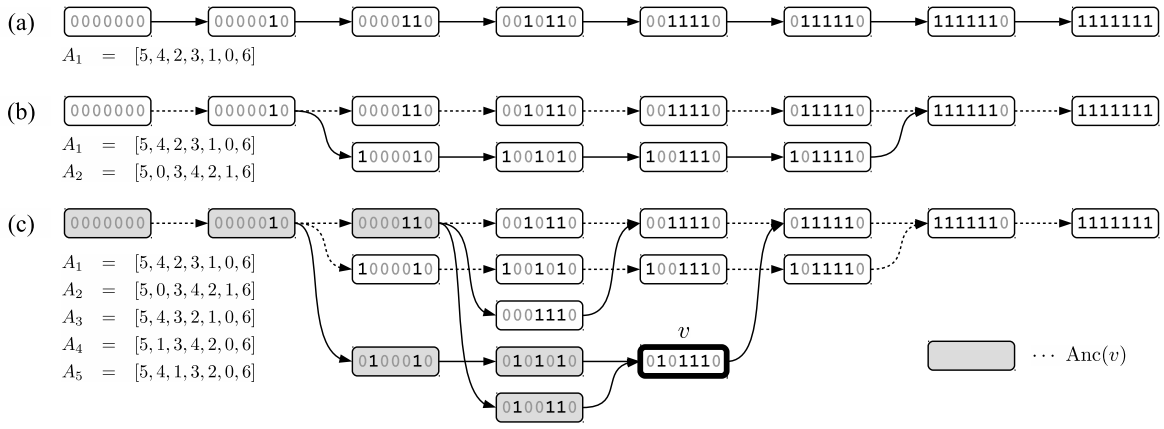


Figure 1: Generating preordering lattice from multiple preordering candidates.

sentences (Schroeder et al., 2009; Jiang et al., 2011); however, the derived confusion network is too constrained to represent variation of preordering, and it cannot take the advantage of alternative reordering in the multiple source sentences.

Compared with above previous works, our method is more generic in that the preordering lattice is constructed based only on the word permutations of the source sentence which are generated from arbitrary and independent preordering methods, and the preordering lattice guarantees that all preordering candidates are compactly encoded in the graph structure. In addition, we show that our preordering lattice approach ourperforms conventional decoding-time reordering methods even with a simple left-to-right dynamic programming algorithm. Our experiments show that the proposed method can achieve comparable or higher translation qualities against a conventional phrase-based method under diverse 11 language pairs: Ar/Zh/Fr/De/It/Ja/Ko/Pt/Ru/Es/Tr into English.

2 Graph Representation of Multiple Preordering Candidates

We denote the source sentence S as an array of words $S = [s_1, s_2, \cdots, s_I]$, and assume that a preordering method takes the source sentence S as an input and returns multiple preordering candidates $\mathcal{A} = \{A_1, A_2, \cdots, A_N\}$, $A_n = [a_1^n, a_2^n, \cdots, a_I^n]$ together with their confidence scores $\mathcal{C} = \{C_1, C_2, \cdots, C_N\} \in \mathbb{R}^N$, where I is the number of words in the source sentence, and N is the number of preordering candidates, and each confidence scores satisfies that $C_j > C_k$ if $j < k$. Each $a_i^n \in \{0, 1, \cdots, I-1\}$ denotes an original position of the source word in S. For example, if we take a Japanese sentence $S =$ [“今日 (today)”, “は”, “いい (good)”, “天気 (weather)”, “です (be)”, “ね”, “。”] (“It is nice weather today.”) and a preordering candidate $A = [5, 4, 2, 3, 1, 0, 6]$, then we obtain a preordered sentence $S' =$ [“ね”, “です”, “いい”, “天気”, “は”, “今日”, “。”].

We introduce an alternative view of the preordering candidate A_n which is represented as a chain graph structure illustrated in Figure 1(a), and we call this graph *preordering lattice*. Each node in the preordering lattice has a *coverage* generated by the preordering candidate, and each edge represents the one-word transition between two coverages. Each coverage represents a set of *processed* words at each timing of decoding, and we encoded each coverage as a bit vector representation in the same way as those used in a conventional phrase-based decoding algorithm. For example, a coverage vector $[0000110]$ indicated that 7 source words should be translated while decoding, and 5th and 6th words are already translated before reaching this coverage. Preordering lattice can be uniquely obtained from a preordering candidate by starting from an empty coverage ($[000\cdots]$) and adding 1 into a_n-th element one-by-one. The process can be regarded as a left-to-right decoding process with single word phrase pairs in a phrase-based decoding.

When multiple preordering candidates are available, we merge them into a single graph structure. Specifically, we integrate nodes in two preordering lattices if their nodes have same coverage vectors each other as shown in Figure 1(b), in which another preordering candidate $[5, 0, 3, 4, 2, 1, 6]$ is merged

Algorithm 1 Integrating preordering lattices

1: $\mathcal{A} \leftarrow$ Array of preordering candidates
2: $G \leftarrow$ Empty lattice
3: **for** $A \in \mathcal{A}$ **do**
4: $\quad G' \leftarrow \mathrm{Lattice}(A)$
5: $\quad$ **for** $v' \in V(G')$ **do**
6: $\quad\quad$ **if** $\neg(\exists v, v \in V(G) \wedge L(v) = L(v'))$ **then**
7: $\quad\quad\quad V(G) \leftarrow V(G) \cup \{v'\}$
8: $\quad\quad$ **end if**
9: $\quad$ **end for**
10: $\quad$ **for** $(v'_1, v'_2) \in E(G')$ **do**
11: $\quad\quad$ **if** $\neg(\exists v_1, v_2, (v_1, v_2) \in E(G) \wedge L(v_1) = L(v'_1) \wedge L(v_2) = L(v'_2))$ **then**
12: $\quad\quad\quad v''_1 \leftarrow v$ s.t. $v \in V(G) \wedge L(v) = L(v'_1)$
13: $\quad\quad\quad v''_2 \leftarrow v$ s.t. $v \in V(G) \wedge L(v) = L(v'_2)$
14: $\quad\quad\quad E(G) \leftarrow E(G) \cup \{(v''_1, v''_2)\}$
15: $\quad\quad$ **end if**
16: $\quad$ **end for**
17: **end for**
18: **return** G

together. In this case, there are 4 nodes with same coverage vectors [0000000], [0000010], [1111110], [1111111] in two preordering lattices, which are integrated as shared nodes. Figure 1(c) shows an example of merging 5 preordering candidates in $\mathcal{A}$ as a single preordering lattice. It is guaranteed that this preordering lattice is uniquely determined given a set of preordering candidates $\mathcal{A}$, and the final graph structure does not depend on the integrating order. Thus, we can merge new lattice paths by enumerating preordering candidates one-by-one. Algorithm 1 shows the method to generate integrated preordering lattice G from $\mathcal{A}$, where $V(G), E(G), E_i(v), E_o(v), L(v)$ represent the sets of nodes/edges in G, input/output edges of given node v, head (exit side) and tail (entrance side) nodes of the edge e, and the label (= coverage vector) of given node v, respectively. $\mathrm{Lattice}(A)$ represents a unique chain graph determined by a preordering candidate A as described above.

Preordeing lattices generated by Algorithm 1 guarantee that all integrated preordering candidates are represented as a path over the lattice, and also guarantee that all words in the source sentence appear only once in any paths over the lattice. In addition, the preordering lattice often includes extra paths which represents other preordering candidates not in the original candidate set $\mathcal{A}$. Thus, the decoding algorithm described in the next section actually explores more preordering candidates when compared with a decoding algorithm which relies only on $\mathcal{A}$.

The preordering lattice is similar to the word lattice structure (Dyer et al., 2008), but all edges in the preordering lattice represent specific words in one source sentence. Daiber et al. (2016) also described a similar structure to our lattice using finite-state transducer (FST), and applied determinization and minimization to compress the lattice. On the other hand, we introduced more simple algorithm described in Algorithm 1 to achieve similar compression.

3 Decoding Algorithm over the Preordering Lattice

We also intoduce a simple decoding algorithm to generate translations directly using the preordering lattice. Our algorithm runs by traversing the lattice in a left-to-right manner. Algorithm 2 presents our decoding algorithm over the preordering lattice without score calculation, where $\mathrm{Begin}(G)$ and $\mathrm{End}(G)$ represents leftmost and rightmost nodes in a given preordering lattice. The decoder determines partial translations at each node in the preordering lattice according to a topological order ($\mathrm{TopologicalSort}(\cdots)$ in line 4), and finally returns the one-best hypothesis at $\mathrm{End}(G)$ ($\mathrm{BestResult}(\cdots)$ in line 15).

Now we focus on the partial translation hypotheses generated at the node v in Figure 1(c). When

Algorithm 2 Decoding over the Preordering Lattice

```
1: G ← Preordering lattice
2: v_L ← Begin(G)
3: H[v_L] ← {""}
4: for v ← TopologicalSort(V(G)\{v_L}) do
5:     H'[v] ← { }
6:     for ∀v'. v' ∈ Anc(v) ∧ IsCandidatePath(v', v) do
7:         for h' ← H[v'] do
8:             for φ ← PhrasePairs(v', v) do
9:                 H'[v] ← H'[v] ∪ {Join(h', φ)}
10:            end for
11:        end for
12:    end for
13:    H[v] ← Prune(H'[v]; K)
14: end for
15: return BestResult(H[End(G)])
```

computing a partial translation hypothesis at v, the decoder first computes ancestor nodes $\mathrm{Anc}(v)$, i.e., a set of nodes from which v is reachable, as shown in the gray nodes in Figure 1(c). Partial hypotheses $H[v'], v' \in \mathrm{Anc}(v)$ is already determined, and the decoder then enumerates new hypotheses $H'[v]$ by concatenating one hypothesis ($\mathrm{Join}(\cdots)$ in line 9) in $H[v']$ and a phrase pair ($\mathrm{PhrasePairs}(\cdots)$ in line 8) connecting v' and v. Note that $H[\cdot]$ and $H'[\cdot]$ may encode additional state information for features requiring non-local contexts, e.g., history of an n-gram language model. Finally, only K top scored hypotheses are preserved in $H[v]$ by applying a beam search strategy ($\mathrm{Prune}(\cdots)$ in line 13).

$\mathrm{IsCandidatePath}(v', v)$ checks whether at least one path exists between v and v' in the original preordering candidates in order to avoid enumerating spurious many phrase pairs.

Each hypothesis $h \in H[v]$ has a score calculated from its feature functions, which are used by $\mathrm{Prune}(\cdots)$ and $\mathrm{BestResult}(\cdots)$ to choose better hypothesis. We used the weighted linear combination for the scoring policy:

$$\mathrm{Score}(h) := \boldsymbol{w}^\top \boldsymbol{f}(h), \tag{1}$$

where $\boldsymbol{w}$ is a weight vector and $\boldsymbol{f}(h)$ is a set of feature functions for each hypothesis h, e.g., features associated with phrase pairs or extra features, such as n-gram language models.

We add scores for each existing path between v' and v in the preordering lattice according to the confidence of preordering candidates, which are used as an additional feature during decoding. After numbers of preliminary experiments, we adopted the product of maximum preordering confidence score and the ratio of phrase length based on our preliminary studies:

$$f(p) := \frac{|p|}{I} \cdot \max_n \gamma(n, p), \tag{2}$$

$$\gamma(n, p) := \begin{cases} C_n, & \text{if } p \subset \mathrm{Lattice}(A_n) \\ -\infty, & \text{otherwise,} \end{cases} \tag{3}$$

where p represents an arbitrary path over the preordering lattice. $-\infty$ means the decoder never choose the path p, and this formulation corresponds to $\mathrm{IsCandidatePath}$ condition in Algorithm 2.

Compared with the conventional decoding method (Zens and Ney, 2008), the proposed method can eliminate some complex score calculations, e.g., rest cost estimation and decoding-time reorderings, because each path in the reordering lattice holds complete information of the word order. As a result, the proposed method makes the decoding algorithm simpler than the conventional method.

4 Experiments

4.1 Experimental settings

We evaluated our proposed method under the settings of translating into English. We chose 11 language pairs consisting of 6 European languages (Fr/De/It/Pt/Ru/Es) and 5 Asian languages (Ar/Zh/Ja/Ko/Tr), which have different linguistic characteristics when compared with English.

For the training data, we used a parallel corpus by mining from the Web using an in-house crawler. The corpus contains 9.5M sentences and 160M words on average, at least 8.0M sentences and 140M words for each language pair. For the development/test data, we separately sampled and manually translated 3,000/5,000 sentences from other data sources on the Web for each language pair. All hyperparameters for each method are optimized using the development data and final evaluation is performed using the test data. During word alignment, IBM Model 1 (Brown et al., 1993) and HMM alignment (Vogel et al., 1996) were performed using one-best preordered source sentences and corresponding target sentences. The phrase table was built according to the alignment results, and shared with all decoding methods. For the English language model, a 4-gram model with stupid backoff smoothing (Brants et al., 2007) was built and commonly used for all settings. Each configuration of the word alignment and the language model was decided according to the preliminary experiments on the baseline system.

For the baseline system, we employed a standard PBMT system, similar to that of (Och and Ney, 2004) with a lexical reordering model (Zens and Ney, 2006) enhanced by a state-of-the-art preordering method based on bracketing transduction grammar (Nakagawa, 2015). We used similar decoding strategy and other basic feature functions to Moses (Koehn et al., 2007) except some neural lexical features such as NNJM (Devlin et al., 2014). Only one-best preordering candidate is used for the baseline system. We chose the best distortion limit of the baseline system for each language pair by the BLEU (Papineni et al., 2002) score on the development data.

We also compared the reranking method (Li et al., 2007), which translates all preordering candidates using conventional PBMT (our baseline system) and chose one with the best score. To do that, we used simple linear interpolation between decoder's score D and preordering confidence C with a hyperparameter λ as follows:

$$\mathrm{Score}(C, D) := \lambda \cdot C + (1 - \lambda) \cdot D. \tag{4}$$

We varied the number of preordering candidates (1, 2, 4, 8, 16, 32, 64-bests) for the proposed method and the reranking method, and chose the one with the best BLEU on the development data. For the reranking method, we trained two variants by differentiating distortion limits, a system sharing the same limit with the PBMT baseline and those with 0, in order to examine the effects of preordering and decoding-time reordering.

For all methods, we used lattice-based Minimum Error-rate Training (MERT) (Macherey et al., 2008) to optimize weights of features.

Evaluation is carried out by BLEU using all test data, and subjective evaluation with 7-grade (0 to 6) Likert scale about translation acceptance using 400 randomly selected samples from the test data in each language pair.

4.2 Results and Discussion

Figure 2 shows the number of nodes in actual preordering lattices generated from each source sentence in Japanese-English test data under 64-best preordering candidates. Upper group in this graph shows the number of unmerged nodes in which nodes are not shared when combining multiple preordering candidates in Algorithm 1, and lower group shows the number of merged nodes. The numbers directly reflect actual computation for decoding. There are averagely 10 times fewer nodes in merged preordering lattices, so our lattice construction of Algorithm 1 efficiently suppress the complexity of decoding when compared with directly using each preordering candidate independently.

Table 1 shows BLEU scores of the PBMT baseline, proposed method, and reranking method, respectively. The table also shows distortion limits (DL) for the PBMT baseline, and numbers of preordering candidates (N) for the proposed method and the reranking method. First, there are roughly the same

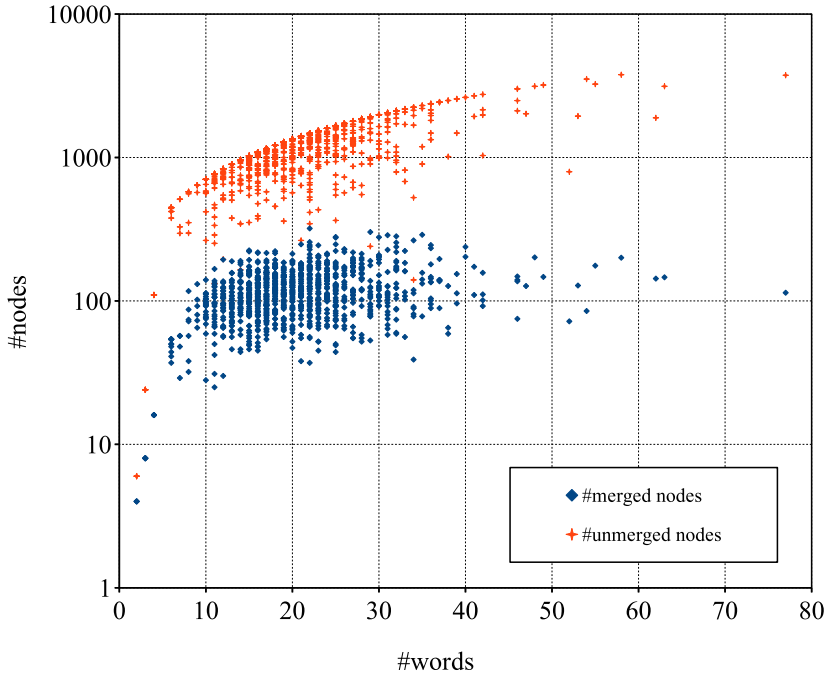


Figure 2: Number of nodes in preordering lattices in Japanese-English translation.

tendencies between the proposed method and the reranking method with similar BLEU improvement, and their systems averagely improves BLEU scores against the PBMT baseline in most language pairs. These results clearly indicate that multiple preordering candidates can largely improve the translation accuracy. In addition, we also see that there are high variance of N mainly in the reranking method. This tendency might come from the accuracy of the preordering method, i.e., if the preordering could perform well, then we require only few preordering candidates to generate accurate translations, and large N introduces less information. Actually, we observed that there is BLEU satulation in some language pairs when using large N, which means low-rank preordering candidates are rarely used to the final translation, according to the language pair.

In comparison between the proposed method and two reranking systems (DL>0 or $=0$), the proposed method without distortion (DL$=0$) often achieves higher BLEU score than DL>0. We conjecture that the tendencies may come from the use of better preordering among multiple candidates instead of a distortion-wise decoding-time reordering. These results clearly show that the decoding-time reordering is not necessary if better reordering is encoded in a preordering lattice.

We also see that there are comparatively higher BLEU improvements when translating from Ja/Ko/Tr than other languages. We speculate that these tendencies come from the grammatical characteristics of source languages. For example, Japanese is one of languages with high flexibility of word order, and the ambiguity may make it difficult to estimate correct preordering. In this case, the use of multiple preordering candidates is a straight-forward way to avoid this problem.

Table 2 shows the results of subjective evaluation for the proposed method against the PBMT baseline. We evaluated statistical significance of each system via t-test of difference between two averages, and this table shows their two-tailed p-values. Note that $\emptyset$ represents some small values ($p < 0.001$). We also included the change rate of these systems, which represents the amount of different translations by both PBMT baseline and the proposed method.

In this table, the proposed method achieves better translations with statistical significance ($p < 0.05$). We can also see that, in 5 Asian to English settings, the proposed method also achieved high translation accuracies under the subjective evaluation, although the proposed method generates divergent translations compared with the PBMT baseline. These languages have more large divergences from English,

Table 1: BLEU scores of each method/language.

Language	PBMT		Proposed			Reranking (DL>0)			Reranking (DL$=0$)		
	BLEU	DL	BLEU	Δ	N	BLEU	Δ	N	BLEU	Δ	N
Ar-En	36.81	6	36.99	+0.18	64	37.28	+0.47	16	36.94	+0.13	32
Zh-En	30.12	4	31.14	+1.02	64	30.90	+0.78	64	31.36	+1.24	64
Fr-En	33.10	5	34.03	+0.93	64	33.98	+0.88	32	34.13	+1.03	64
De-En	30.36	6	31.05	+0.69	32	31.53	+1.17	16	31.35	+0.99	32
It-En	37.65	6	38.22	+0.57	64	38.70	+1.05	16	38.40	+0.75	8
Ja-En	15.51	5	16.68	+1.17	32	17.01	+1.50	64	16.58	+1.07	32
Ko-En	20.32	5	22.34	+2.02	64	22.75	+2.43	64	22.86	+2.54	64
Pt-En	40.66	6	41.43	+0.77	64	41.48	+0.82	64	41.38	+0.72	64
Ru-En	25.45	6	25.79	+0.34	64	26.12	+0.67	16	25.55	+0.10	32
Es-En	35.50	5	36.11	+0.61	64	34.96	−0.54	8	36.42	+0.92	64
Tr-En	26.71	6	28.87	+2.16	64	29.27	+2.56	32	29.05	+2.34	64

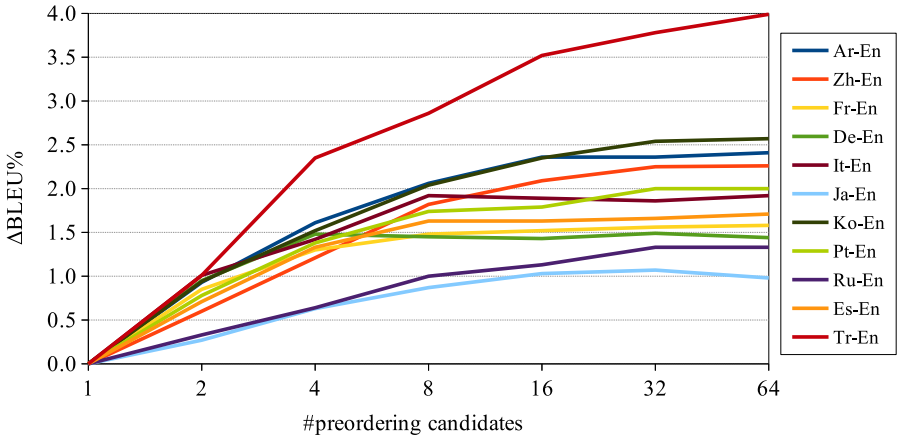
Table 2: Results of subjective evaluation.

Language	Score (PBMT)	Score (Proposed)	Δ	p-value	Change rate%
Ar-En	3.998	4.128	+0.130	0.028	66.14
Zh-En	3.135	3.278	+0.143	0.024	77.94
Fr-En	4.278	4.563	+0.285	$\emptyset$	36.81
De-En	3.902	4.259	+0.358	$\emptyset$	39.80
It-En	4.286	4.429	+0.143	0.046	35.54
Ja-En	2.943	3.238	+0.295	$\emptyset$	80.33
Ko-En	2.900	3.139	+0.239	$\emptyset$	70.44
Pt-En	4.392	4.642	+0.250	0.004	32.64
Ru-En	4.003	4.160	+0.158	0.029	41.61
Es-En	4.237	4.262	+0.025	0.712	48.78
Tr-En	3.150	3.553	+0.403	$\emptyset$	79.18

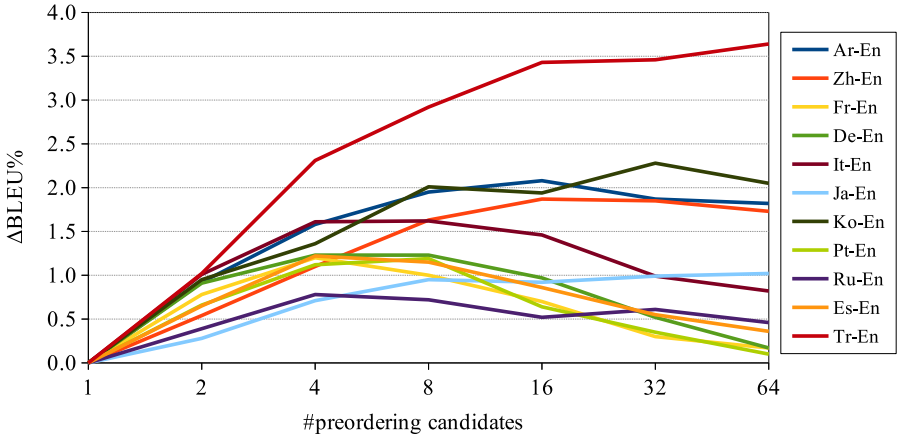
and estimating correct preorderings is more difficult than European languages. By these results, we can also say that the proposed method performs more effectively than PBMT baseline under BLEU and subjective evaluation.

Figure 3 shows BLEU changes of the proposed method by increasing the number of preordering candidates N. The baselines of these graphs are the BLEU score using only one-best preordering candidate, and these scores are roughly similar to a conventional PBMT system with DL$=0$. Figure 3(a) shows cases that use preordering confidence scores described in Section 2 as an additional feature, and Figure 3(b) shows cases of ignoring those scores. In Figure 3(a), the proposed method improves translation accuracy by increasing the number of preordering candidates in nearly all language pairs. Figure 3(a) also shows the BLEU satulation when using large N described in the previous paragraph. And in Figure 3(b), there are non-negligible BLEU reduction by using many preordering candidates. This tendency is expected, because ignoring confidence scores of preordering candidates implies treating all preordering candidates with the same importance, and the decoder have finally chosen the hypothesis with accidentally high scores by other features. Thus, introducing preordering confidence into decoding features is effective to prevent these kind of errors and guarantee translation accuracies.

Figure 4 shows mean decoding times of the PBMT baseline and the proposed method in Japanese-English setting with various decoding parameters. In the PBMT baseline, we changed both distortion limit (0 to 6) and beam width for each coverage of source sentence (1, 2, 4, 8,16 ,32 ,64 ,128). In the proposed method, we varied the number of preordering candidates (1, 2, 4, 8, 16, 32, 64) and beam width as same as PBMT baseline. Basically, increasing distortion limit or the number of preordering candi-



(a) With path score



(b) Without path score

Figure 3: BLEU changes according to the number of applied preordering candidates.

dates require much computation amount. In this figure, the proposed method using many preordering candidates can achieve high translation accuracy, as well as the proposed method runs as same range of computation time as PBMT.

Table 3 shows some examples in Japanese-English setting. In the first and second examples, the proposed method achieves better translation by exploiting multiple candidates. However, the last example demonstrates a weakness in our method mainly caused by the low confidence in preordering decision, e.g., parallel phrases. In this case, the language model of "image information" is stronger than that of "information and images" and this difference of scores exceeds the preordering confidence. As a result, the decoder fails to choose correct preordering. Avoiding these kinds of problems should be one of our future work.

5 Conclusion

In this paper, we proposed a new phrase-based decoding method using multiple preordering candidates. Our method outperforms previous PBMT systems without using any decoding-time reordering.

In this study, we used only one preordering method. Our method can be easily extended to any preordering methods along as they can emit N-best preordering candidates with optional confidence scores. In addition, the proposed method further may be able to combine multiple preordering candidates from different preordering methods by introducing multiple path scores for each preordering methods. In future work, we will plan to evaluate the effect of using or combining other preordering methods for the proposed method.

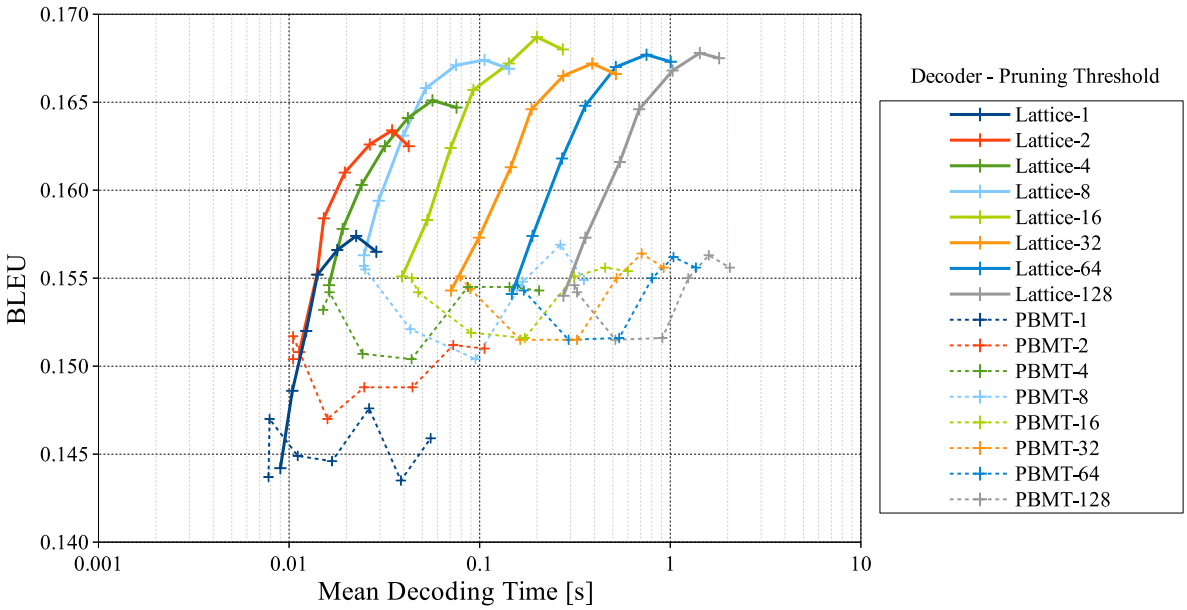


Figure 4: Relationship between decoding time and BLEU in Japanese-English translation.

Table 3: Translate examples of PBMT baseline and proposed method.

Type	Source Sentence/Translate	Score
Source	では, この問題をどうやって解決するつもりですか.	
PBMT	So, are you going to solve **how** this problem.	1
Proposed	So, **how** do you intend to solve this problem.	6
Source	私の車は, 私を含む全員がシートベルトを着用するまで駆動しません.	
PBMT	My car, **everyone including the I does not drive up** to wear a seat belt.	1
Proposed	My car **does not drive until everyone, including me** to wear a seat belt.	5
Source	技術革新により, 情報と画像をカードの表面に印刷できます.	
PBMT	By technological innovation, you can print the **information and images** on the card surface.	6
Proposed	By technological innovation, you can print the **image information** on the surface of the card.	4

Acknowledgement

Part of this work was supported by Grant-in-Aid for JSPS Fellows Grant Number 15J10649.

References

Thorsten Brants, Ashok C Popat, Peng Xu, Franz J Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proc. EMNLP-CoNLL*, pages 858–867.

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

Joachim Daiber, Miloš Stanojevic, Wilker Aziz, and Khalil Sima ' an. 2016. Examining the relationship between preordering and word order freedom in machine translation. In *Proc. WMT*, pages 118–130.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proc. ACL*, pages 1370–1380.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *Proc. ACL-HLT*, pages 1012–1020.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proc. EMNLP*, pages 848–856.

Teresa Herrmann, Jan Niehues, and Alex Waibel. 2013a. Combining word reordering methods on different linguistic abstraction levels for statistical machine translation. In *Proc. SSST*, pages 39–47.

Teresa Herrmann, Jochen Weiner, Jan Niehues, and Alex Waibel. 2013b. Analyzing the potential of source sentence reordering in statistical machine translation. In *Proc. IWSLT*.

Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2010. Head finalization: A simple reordering rule for SOV languages. In *Proc. WMT-MetricsMATR*, pages 244–251.

Jie Jiang, Jinhua Du, and Andy Way. 2011. Incorporating source-language paraphrases into phrase-based smt with confusion networks. In *Proc. SSST*, pages 31–40.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. NAACL-HLT*, pages 48–54.

Philipp Koehn, Amittai Axelrod, Alexandra Birch, Chris Callison-Burch, Miles Osborne, David Talbot, and Michael White. 2005. Edinburgh system description for the 2005 iwslt speech translation evaluation. In *Proc. IWSLT*, pages 68–75.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. ACL*, pages 177–180.

Chi-Ho Li, Minghui Li, Dongdong Zhang, Mu Li, Ming Zhou, and Yi Guan. 2007. A probabilistic approach to syntax-based reordering for statistical machine translation. In *Proc. ACL*, pages 720–727.

Wolfgang Macherey, Franz Josef Och, Ignacio Thayer, and Jakob Uszkoreit. 2008. Lattice-based minimum error rate training for statistical machine translation. In *Proc. EMNLP*, pages 725–734.

Tetsuji Nakagawa. 2015. Efficient top-down btg parsing for machine translation preordering. In *Proc. ACL-IJCNLP*, pages 208–218.

Graham Neubig, Taro Watanabe, and Shinsuke Mori. 2012. Inducing a discriminative parser to optimize machine translation reordering. In *Proc. EMNLP-CoNLL*, pages 843–853.

Jan Niehues and Muntsin Kolss. 2009. A POS-based model for long-range reorderings in SMT. In *Proc. WMT*, pages 206–214.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proc. ACL*, pages 311–318.

Josh Schroeder, Trevor Cohn, and Philipp Koehn. 2009. Word lattices for multi-source translation. In *Proc. EACL*, pages 719–727.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proc. COLING*, pages 836–841.

Fei Xia and Michael McCord. 2004. Improving a statistical mt system with automatically learned rewrite patterns. In *Proc. COLING*, pages 508–514.

Richard Zens and Hermann Ney. 2006. Discriminative reordering models for statistical machine translation. In *Proc. WMT*, pages 55–63.

Richard Zens and Hermann Ney. 2008. Improvements in dynamic programming beam search for phrase-based statistical machine translation. In *Proc. IWSLT*, pages 198–205.

Zhongyuan Zhu. 2014. Weblio pre-reordering statistical machine translation system. In *Proc. WAT*, pages 33–38.