Phrase Table Pruning via Submodular Function Maximization

Masaaki Nishino and **Jun Suzuki** and **Masaaki Nagata**
NTT Communication Science Laboratories, NTT Corporation
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237, Japan
{nishino.masaaki,suzuki.jun,nagata.masaaki}@lab.ntt.co.jp

Abstract

Phrase table pruning is the act of removing phrase pairs from a phrase table to make it smaller, ideally removing the least useful phrases first. We propose a phrase table pruning method that formulates the task as a submodular function maximization problem, and solves it by using a greedy heuristic algorithm. The proposed method can scale with input size and long phrases, and experiments show that it achieves higher BLEU scores than state-of-the-art pruning methods.

1 Introduction

A phrase table, a key component of phrase-based statistical machine translation (PBMT) systems, consists of a set of phrase pairs. A phrase pair is a pair of source and target language phrases, and is used as the atomic translation unit. Today's PBMT systems have to store and process large phrase tables that contain more than 100M phrase pairs, and their sheer size prevents PBMT systems for running in resource-limited environments such as mobile phones. Even if a computer has enough resources, the large phrase tables increase turnaround time and prevent the rapid development of MT systems.

Phrase table pruning is the technique of removing ineffective phrase pairs from a phrase table to make it smaller while minimizing the performance degradation. Existing phrase table pruning methods use different metrics to rank the phrase pairs contained in the table, and then remove low-ranked pairs. Metrics used in previous work are frequency, conditional probability, and Fisher's exact test score (Johnson et al., 2007). Zens et al. (2012) evaluated many phrase table pruning methods, and concluded that entropy-based pruning method (Ling et al., 2012; Zens et al., 2012) offers the best performance. The entropy-based pruning method uses entropy to measure the redundancy of a phrase pair, where we say a phrase pair is redundant if it can be replaced by other phrase pairs. The entropy-based pruning method runs in time linear to the number of phrase-pairs. Unfortunately, its running time is also exponential to the length of phrases contained in the phrase pairs, since it contains the problem of finding an optimal phrase alignment, which is known to be NP-hard (DeNero and Klein, 2008). Therefore, the method can be impractical if the phrase pairs consist of longer phrases.

In this paper, we introduce a novel phrase table pruning method that formulates and solves the phrase table pruning problem as a submodular function maximization problem. A submodular function is a kind of set function that satisfies the submodularity property. Generally, the submodular function maximization problem is NP-hard, however, it is known that $(1 - 1/e)$ optimal solutions can be obtained by using a simple greedy algorithm (Nemhauser et al., 1978). Since a greedy algorithm scales with large inputs, our method can be applicable to large phrase tables.

One key factor of the proposed method is its carefully designed objective function that evaluates the quality of a given phrase table. In this paper, we use a simple monotone submodular function that evaluates the quality of a given phrase table by its coverage of a training corpus. Our method is simple, parameter free, and does not cause exponential explosion of the computation time with longer phrases. We conduct experiments with two different language pairs, and show that the proposed method shows higher BLEU scores than state-of-the-art pruning methods.

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 406–411,
Berlin, Germany, August 7-12, 2016.

2 Submodular Function Maximization

Let Ω be a base set consisting of M elements, and $g : 2^{\Omega} \mapsto \mathbb{R}$ be a set function that upon the input of $X \subseteq \Omega$ returns a real value. If g is a submodular function, then it satisfies the condition

$$g(X \cup \{x\}) - g(X) \geq g(Y \cup \{x\}) - g(Y)\,,$$

where $X, Y \in 2^{\Omega}$, $X \subseteq Y$, and $x \in \Omega \setminus Y$. This condition represents the diminishing return property of a submodular function, i.e., the increase in the value of the function due to the addition of item x to Y is always smaller than that obtained by adding x to any subset $X \subseteq Y$. We say a submodular function is *monotone* if $g(Y) \geq g(X)$ for any $X, Y \in 2^{\Omega}$ satisfying $X \subseteq Y$. Since a submodular function has many useful properties, it appears in a wide range of applications (Kempe et al., 2003; Lin and Bilmes, 2010; Kirchhoff and Bilmes, 2014).

The maximization problem of a monotone submodular function under cardinality constraints is formulated as

$$\begin{aligned} &\text{Maximize } g(X) \\ &\text{Subject to } X \in 2^{\Omega} \text{ and } |X| \leq K\,, \end{aligned}$$

where $g(X)$ is a monotone submodular function and K is the parameter that defines maximum cardinality. This problem is known to be NP-hard, but a greedy algorithm can find an approximate solution whose score is certified to be $(1 - 1/e)$ optimal (Nemhauser et al., 1978). Algorithm 1 shows a greedy approximation method the can solve the submodular function maximization problem under cardinality constraints. This algorithm first sets $X \leftarrow \emptyset$, and adds item $x^* \in \Omega \setminus X$ that maximizes $g(X \cup \{x^*\}) - g(X)$ to X until $|X| = K$.

Assuming that the evaluation of $g(X)$ can be performed in constant time, the running time of the greedy algorithm is $O(MK)$ because we need $O(M)$ evaluations of $g(X)$ for selecting x^* that maximizes $g(X \cup \{x^*\}) - g(X)$, and these evaluations are repeated K times. If we naively apply the algorithm to situations where M is very large, then the algorithm may not work in reasonable running time. However, an accelerated greedy algorithm can work with large inputs (Minoux, 1978; Leskovec et al., 2007), since it can drastically reduce the number of function evaluations from MK. We applied the accelerated greedy algorithm in the following experiments, and found it could solve the problems in 24 hours. Moreover, further enhancement can be achieved by applying distributed algorithms (Mirzasoleiman et al., 2013) and stochastic greedy algorithms (Mirzasoleiman et al., 2015).

Algorithm 1 Greedy algorithm for maximizing a submodular function

Input: Base set Ω, cardinality K
Output: $X \in 2^{\Omega}$ satisfying $|X| = K$.

```
X ← ∅
while |X| < K do
    x* ← arg max_{x∈Ω\X} g(X ∪ {x}) − g(X)
    X ← X ∪ {x*}
output X
```

3 Phrase Table Pruning

We first define some notations. Let $\Omega = \{x_1, \ldots, x_M\}$ be a phrase table that has M phrase pairs. Each phrase pair, x_i, consists of a source language phrase, p_i, and a target language phrase, q_i, and is written as $x_i = \langle p_i, q_i \rangle$. Phrases p_i and q_i are sequences of words $p_i = (p_{i1}, \ldots, p_{i|p_i|})$ and $q_i = (q_{i1}, \ldots, q_{i|q_i|})$, where p_{ij} represents the j-th word of p_i and q_{ij} represents the j-th word of q_i. Let t_i be the i-th translation pair contained in the training corpus, namely $t_i = \langle f_i, e_i \rangle$, where f_i and e_i are source and target sentences, respectively. Let N be the number of translation pairs contained in the corpus. f_i and e_i are represented as sequences of words $f_i = (f_{i1}, \ldots, f_{i|f_i|})$ and $e_i = (e_{i1}, \ldots, e_{i|e_i|})$, where f_{ij} is the j-th word of sentence f_i and e_{ij} is the j-th word of sentence e_i.

Definition 1. Let $x_j = \langle p_j, q_j \rangle$ be a phrase pair and $t_i = \langle f_i, e_i \rangle$ be a translation pair. We say x_j *appears* in t_i if p_j is contained in f_i as a subsequence and q_j is contained in e_i as a subsequence. We say phrase pair x_j *covers* word f_{ik} if x_j appears in $\langle f_i, e_i \rangle$ and f_{ik} is contained in the subsequence that equals p_j. Similarly, we say x_j covers e_{ik} if x_j appears in $\langle f_i, e_i \rangle$ and e_{ik} is contained in the subsequence that equals q_j.

Using the above definitions, we describe here our phrase-table pruning algorithm; it formulates the task as a combinatorial optimization problem. Since phrase table pruning is the problem of finding a subset of Ω, we formulate the problem as a submodular function maximization problem under cardinality constraints, i.e., the problem is finding

$X \subseteq \Omega$ that maximizes objective function $g(X)$ while satisfying the condition $|X| = K$, where K is the size of pruned phrase table. If $g(X)$ is a monotone submodular function, we can apply Algorithm 1 to obtain an $(1 - 1/e)$ approximate solution. We use the following objective function.

$$g(X) = \sum_{i=1}^{N} \sum_{k=1}^{|f_i|} \log\left[c(X, f_{ik}) + 1\right] + \sum_{i=1}^{N} \sum_{k=1}^{|e_i|} \log\left[c(X, e_{ik}) + 1\right],$$

where $c(X, f_{ik})$ is the number of phrase pairs contained in X that cover f_{ik}, the k-th word of the i-th source sentence f_i. Similarly, $c(X, e_{ik})$ is the number of phrase pairs that cover e_{ik}.

Example 1. Consider phrase table X holding phrase pairs $x_1 = \langle(\text{das Haus}), (\text{the house})\rangle$, $x_2 = \langle(\text{Haus}), (\text{house})\rangle$, and $x_3 = \langle(\text{das Haus}), (\text{the building})\rangle$. If a corpus consists of a pair of sentences $f_1 =$ "das Haus ist klein" and $e_1 =$ "this house is small", then x_1 and x_2 appear in $\langle f_1, e_1\rangle$ and word f_{12} = "Haus" is covered by x_1 and x_2. Hence $c(X, f_{12}) = 2$.

This objective function basically gives high scores to X if it contains many words of the training corpus. However, since we take the logarithm of cover counts $c(X, f_{ik})$ and $c(X, e_{ik})$, $g(X)$ becomes high when X covers many different words. This objective function prefers to select phrase pairs that frequently appear in the training corpus but with low redundantly. This objective function prefers pruned phrase table X that contains phrase pairs that frequently appear in the training corpus, with no redundant phrase pairs. We prove the submodularity of the objective function below.

Proposition 1. *$g(X)$ is a monotone submodular function.*

Proof. Apparently, every $c(X, f_{ik})$ and $c(X, e_{ik})$ is a monotone function of X, and it satisfies the diminishing return property since $c(X \cup \{x\}, f_{ik}) - c(X, f_{ik}) = c(Y \cup \{x\}, f_{ik}) - c(Y, f_{ik})$ for any $X \subseteq Y$ and $x \notin Y$. If function $h(X)$ is monotone and submodular, then $\phi(h(X))$ is also monotone and submodular for any concave function $\phi : \mathbb{R} \mapsto \mathbb{R}$. Since $\log(X)$ is concave, every $\log[c(X, f_{ik})+1]$ and $\log[c(X, e_{ik})+1]$ is a monotone submodular function. Finally, if $h_1, \ldots, h_n$ are monotone and submodular, then $\sum_i h_i$ is also monotone and submodular. Thus $g(X)$ is monotone and submodular. □

Computation costs If we know all counts $c(X, f_{ik})$ and $c(X, e_{ik})$ for all f_{ik}, e_{ik}, then $g(X \cup \{x\})$ can be evaluated in time linear with the number of words contained in the training corpus[1]. Thus our algorithm does not cause exponential explosion of the computation time with longer phrases.

4 Evaluation

4.1 Settings

We conducted experiments on the Chinese-English and Arabic-English datasets used in NIST OpenMT 2012. In each experiment, English was set as the target language. We used Moses (Koehn et al., 2007) as the phrase-based machine translation system. We used the 5-gram Kneser-Ney language model trained separately using the English GigaWord V5 corpus (LDC2011T07), a monolingual corpus distributed at WMT 2012, and Google Web 1T 5-gram data (LDC2006T13). Word alignments are obtained by running giza++ (Och and Ney, 2003) included in the Moses system. As the test data, we used 1378 segments for the Arabic-English dataset and 2190 segments for the Chinese-English dataset, where all test segments have 4 references (LDC2013T07, LDC2013T03). The tuning set consists of about 5000 segments gathered from MT02 to MT06 evaluation sets (LDC2010T10, LDC2010T11, LDC2010T12, LDC2010T14, LDC2010T17). We set the maximum length of extracted phrases to 7. Table 1 shows the sizes of phrase tables. Following the settings used in (Zens et al., 2012), we reduce the effects of other components by using the same feature weights obtained by running the MERT training algorithm (Och, 2003) on full size phrase tables and tuning data to all pruned tables. We run MERT for 10 times to obtain 10 different feature weights. The BLEU scores reported in the following experiments are the averages of the results obtained by using these different feature weights.

We adopt the entropy-based pruning method used in (Ling et al., 2012; Zens et al., 2012) as the baseline method, since it shows best BLEU

[1] Running time can be further reduced if we compute the set of words covered by each phrase pair x_i before executing the greedy algorithm.

Language Pair	Number of phrase pairs
Arabic-English	234M
Chinese-English	169M

Table 1: Phrase table sizes.

scores as per (Zens et al., 2012). We used the parameter value of the entropy-based method suggested in (Zens et al., 2012). We also compared with the significance-based method (Johnson et al., 2007), which uses Fisher's exact test to calculate significance scores of phrase pairs and prunes less-significant phrase pairs.

4.2 Results

Figure 1 and Figure 2 show the BLEU scores of pruned tables. The horizontal axis is the number of phrase pairs contained in a table, and the vertical axis is the BLEU score. The values in the figure are difference of BLEU scores between the proposed method and the baseline method that shows higher score. In the experiment with the Arabic-English dataset, both methods can remove 80% of phrase pairs without losing 1 BLEU point, and the proposed method shows better performance than the baseline methods for all table sizes. The difference in BLEU scores becomes larger when table sizes are small. In the experiment on the Chinese-English dataset, both methods can remove 80% of phrase pairs without losing 1 BLEU point, and the proposed method also shows comparable or better performance. The difference in BLEU scores also becomes larger when table sizes are small.

Figure 3 shows phrase table sizes in the binarized and compressed phrase table format used in Moses (Junczys-Dowmunt, 2012). The horizontal axis is the number of phrase pairs contained in the table, and the vertical axis is phrase table size. We can see that there is a linear relationship between phrase table sizes and the number of phrase pairs. The original phrase table requires 2.8GB memory. In contrast, the 90% pruned table only requires 350MB of memory. This result shows the effectiveness of phrase table pruning on reducing resource requirements in practical situations.

5 Related Work

Previous phrase table pruning methods fall into two groups. Self-contained methods only use resources already used in the MT system, e.g., training corpus and phrase tables. Entropy-based

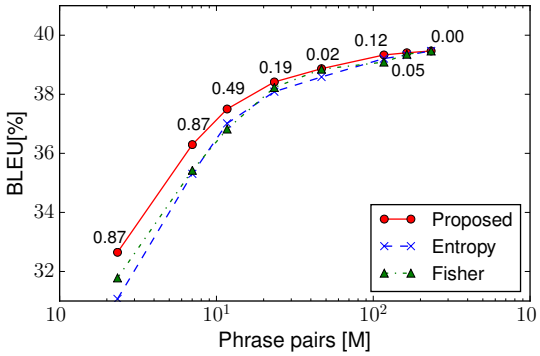

Figure 1: BLEU score as a function of the number of phrase pairs (Arabic-English).

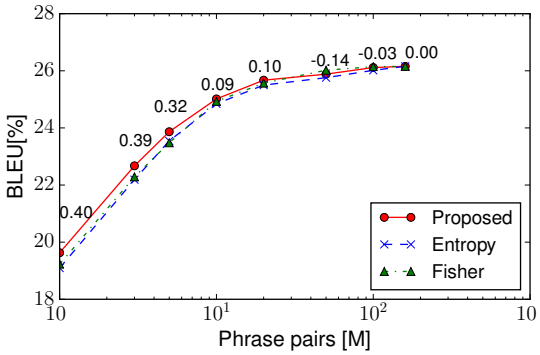

Figure 2: BLEU score as a function of the number of phrase pairs (Chinese-English).

methods (Ling et al., 2012; Zens et al., 2012), a significance-based method (Johnson et al., 2007), and our method are self-contained methods. Non self-contained methods exploit usage statistics for phrase pairs (Eck et al., 2007) and additional bilingual corpora (Chen et al., 2009). Since self contained methods require additional resources, it is easy to apply to existing MT systems.

Effectiveness of the submodular functions maximization formulation is confirmed in various NLP applications including text summarization (Lin and Bilmes, 2010; Lin and Bilmes, 2011) and training data selection for machine translation (Kirchhoff and Bilmes, 2014). These methods are used for selecting a subset that contains important items but not redundant items. This paper can be seen as applying the subset selection formulation to the phrase table pruning problem.

6 Conclusion

We have introduced a method that solves the phrase table pruning problem as a submodular function maximization problem under cardinal-

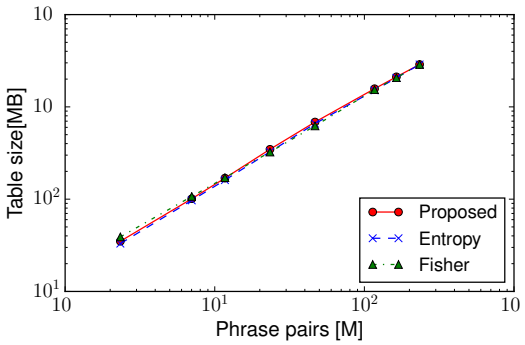

Figure 3: Moses compact phrase table size as a function of the number of phrase pairs (Arabic-English).

ity constraints. Finding an optimal solution of the problem is NP-hard, so we apply a scalable greedy heuristic to find $(1 - 1/e)$ optimal solutions. Experiments showed that our greedy algorithm, which uses a relatively simple objective function, can achieve better performance than state-of-the-art pruning methods.

Our proposed method can be easily extended by using other types of submodular functions. The objective function used in this paper is a simple one, but it is easily enhanced by the addition of metrics used in existing phrase table pruning techniques, such as Fisher's exact test scores and entropy scores. Testing such kinds of objective function enhancements is an important future task.

References

Yu Chen, Martin Kay, and Andreas Eisele. 2009. Intersecting multilingual data for faster and better statistical translations. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 128–136.

John DeNero and Dan Klein. 2008. The complexity of phrase alignment problems. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 25–28.

Matthias Eck, Stephan Vogel, and Alex Waibel. 2007. Translation model pruning via usage statistics for statistical machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 21–24.

Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. 2007. Improving translation quality by discarding most of the phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 967–975.

Marcin Junczys-Dowmunt. 2012. Phrasal rank-encoding: Exploiting phrase redundancy and translational relations for phrase table compression. *The Prague Bulletin of Mathematical Linguistics*, 98:63–74.

David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pages 137–146.

Katrin Kirchhoff and Jeff Bilmes. 2014. Submodularity for data selection in machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 131–141.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180.

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 420–429.

Hui Lin and Jeff Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 912–920.

Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 510–520.

Wang Ling, João Graça, Isabel Trancoso, and Alan Black. 2012. Entropy-based pruning for phrase-based machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 962–971.

Michel Minoux. 1978. Accelerated greedy algorithms for maximizing submodular set functions. In *Proceedings of the 8th IFIP Conference on Optimization Techniques*, pages 234–243.

Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. 2013. Distributed submodular maximization: Identifying representative elements in massive data. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2049–2057.

Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, Amin Karbasi, Jan Vondrák, and Andreas Krause. 2015. Lazier than lazy greedy. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1812–1818.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294.

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

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167.

Richard Zens, Daisy Stanton, and Peng Xu. 2012. A systematic comparison of phrase table pruning techniques. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 972–983.