Making Language Model as Small as Possible in Statistical Machine Translation

Yang Liu[1], Jiajun Zhang[1], Jie Hao[2], and Dakun Zhang[2]

[1] NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] Toshiba (China) R&D Center, Beijing, China
{yang.liu2013,jjzhang}@nlpr.ia.ac.cn
{haojie,zhangdakun}@toshiba.com.cn

Abstract. As one of the key components, n-gram language model is most frequently used in statistical machine translation. Typically, higher order of the language model leads to better translation performance. However, higher order of the n-gram language model requires much more monolingual training data to avoid data sparseness. Furthermore, the model size increases exponentially when the n-gram order becomes higher and higher. In this paper, we investigate the language model pruning techniques that aim at making the model size as small as possible while keeping the translation quality. According to our investigation, we further propose to replace the higher order n-grams with a low-order cluster-based language model. The extensive experiments show that our method is very effective.

Keywords: language model pruning, frequent n-gram clustering, statistical machine translation.

1 Introduction

In statistical machine translation (SMT), language model is one of the key components (others include translation model and reordering model). Typically, SMT applies a word-based n-gram language model which predicts the nth word using the previous n-1 words as the context. Although the n-gram language model ignores the intrinsic structure of the natural language, it is employed in almost all of the SMT models due to its simplicity and effectiveness.

Furthermore, it is well acknowledged that the higher the n-gram order (not too high), the better the SMT performance is. However, in order to avoid the data sparseness problem, higher n-gram language model needs much more training data for parameter estimation. Moreover, the size of the model increases exponentially with the order grows. In practical use, we usually require the language model size to be as small as possible while keeping its capacity as much as possible. Accordingly, many language model pruning techniques are proposed, including count-cutoffs (Jelinek et al., 1990), weighted difference pruning (and its variants) (Seymore and Rosenfeld, 1996; Moore and Quirk, 2009), Stolcke pruning (Stolcke, 1998), and IBM clustering (Brown et al, 1990). The previous works show that these pruning approaches perform

X. Shi and Y. Chen (Eds.): CWMT 2014, CCIS 493, pp. 1–12, 2014.

similarly with respect to the language model perplexity. But, few studies investigate the relationship between the pruning techniques (language model size) and the SMT performance.

In this paper, we aim at conducting a comprehensive investigation to figure out how the SMT performance is influenced by the language model pruning technique. For simplicity and efficiency, we just adopt the count-cutoff pruning technique with modified Knerser-Ney discounting. In this study, we investigate various pruning options and evaluate how they influence the translation quality of different SMT models (the conventional phrases-based model and the formal syntax-based model).

According to the deep analysis of the investigation results, we find an interesting phenomenon: many 4-gram and 5-gram instances share the same frequent trigram. Based on this finding, we propose a simple but effective approach that makes full use of the frequent trigrams and their contexts to replace all the 4-gram and 5-gram instances. We further propose to cluster the frequent trigrams with Brown Clustering and re-train a cluster-based trigram language model. By doing so, we hope the original trigram language model plus the new cluster-based trigram one can obtain the similar translation performance while retaining a small model size.

The extensive large-scale experiments demonstrate that the pruning technique can discard about half of the ngrams for the 5-gram language model while keeping the same translation performance as the original 5-gram language model. By clustering the frequent trigrams and retraining a new cluster-based trigram language model, we can keep the language model size as small as the original trigram language model, but can improve the translation quality over the original trigram one statistical significantly, although it still underperforms the 5-gram language model.

2 Language Model Pruning for Machine Translation

2.1 Language Model Pruning

In classical language modeling, the language model pruning aims to reduce the language model size as much as possible, as long as the language model perplexity does not increase. Conventionally, there are four well-known pruning techniques including count-cutoffs (Jelinek et al., 1990), weighted difference pruning (and its variants) (Seymore and Rosenfeld, 1996; Moore and Quirk, 2009), Stolcke pruning (Stolcke, 1998), and IBM clustering (Brown et al, 1990). As the n-gram model is widely adopted, the model pruning techniques usually attempt to remove the n-gram entries as many as possible.

For example, in the count-cutoffs pruning technique (Jelinek et al., 1990), any n-gram with the occurrence times below the pre-defined cutoff threshold will be discarded. This method can result in significantly small models, with slight increase in perplexity.

The weighted difference method (Seymore and Rosenfeld, 1996) focuses on the difference between the previous words. Considering the phrase “the Great Wall”, if the bigram probability $P(Wall|Great)$ is the same as the trigram probability $P(Wall|the\ Great)$, the trigram probability $P(Wall|the\ Great)$ is unnecessary and the trigram entry “the Great Wall” will be discarded.

Stolck pruning (Stolcke, 1998) is a pruning approach which is based on relative entropy. Stolcke first calculate the relative entropy as follows:

$$\sum_{x,y,z} p(xyz)[logP'(z|xy) - logP(z|xy)] \quad (1)$$

Here, *P'* denotes the model after pruning, and *P* denotes the original model. Trigram probability *P*(*z*|*xy*) will increase the entropy of the model. If the entropy increased by the particular trigram less than the pruning threshold, this trigram will be removed. It is shown that this method has similar result with the weighted difference method in practice.

IBM clustering (or Brown clustering) (Brown et al., 1990) is a word-based clustering technique. The idea of IBM clustering is that similar words always appear in similar context. Thus, the cluster of those words, called word class, can be used as a variable to substitute the original word and estimate the corresponding probability.

Goodman and Gao (2000) has conducted a comprehensive comparison for these four techniques in the purely language modeling. The results imply that although count-cutoff method is not the best choice in some cases, the difference on reduced size and perplexity between those pruning become negligible with increasing training data. Moreover, in Moore and Quirk's (2009) work, it is shown that the simple count cutoff method with modified Kneser-Ney discounting smoothing technique can achieve similar result with weighted-difference variant method.

From the pruning techniques in purely language modeling, count cutoff is a simple but effective approach. As few researchers study the relationship between the language model size and the translation quality, we choose the count cutoff technique with modified Kneser-Ney discounting smoothing to prune the language model in the machine translation scenario.

2.2 Language Model Pruning for The Phrase-Based Translation

In practical applications, the phrase-based model is the most popular model to construct the translation system due to its simplicity, effectiveness and efficiency. The phrase-based translation model generates the output from left to right by expanding one phrase translation each time. As one of the most key components, the language model attempts to distinguish the fluent translations from the ungrammatical outputs.

To prune the language model with the count cutoff technique, the only parameter is the count cutoff threshold. As mentioned in the previous section, any n-gram whose occurrence is below this threshold will be discarded. It should be noted that the count cutoff based n-gram pruning does not only affect the n-gram itself, but also influence the low order n-grams during the language model training.

We conduct our experiments on Chinese-to-English translation, using the state-of-the-art phrase-based system Moses (Koehn et al., 2007).We perform word alignment with GIZA++ (Och, 2000). The word alignments are symmetrized using the grow-diag-final-and heuristic. The bilingual training corpus comes from LDC[1], including

[1] LDC category numbers: LDC2000T50, LDC2002L27, LDC2003E07, LDC2003E14, LDC2004T07, LDC2005T06, LDC2005T10 and LDC2005T34.

about 2.1 million parallel sentence pairs. We apply the SRILM toolkit[2] (SRI Language Modeling Toolkit) to train the baseline n-gram language model with the modified Knerser-Ney discount smoothing using the target side of the bilingual training data plus the Xinhua portion of the English Gigaword corpus. To prune the baseline language model, we adopt the option "-gt*n*min" in the SRILM toolkit to investigate the relations between the language model size and the translation performance.

We use the NIST MT2003 test data as the development set, and NIST MT2004 as the test set. The evaluation metric is case-insensitive BLEU-4 with shortest length penalty.

Table 1 shows the statistics of the language model size and the translation quality using the baseline language model with n-gram order from 1 to 7. Each n-gram model is trained using the same modified Knerser-Ney discount smoothing technique.

Table 1. Performance of the phrase-based translation with each n-gram language model

Language Model	Size(MB)	BLEU	Perplexity
Unigram	32	22.31	1155.74
Bigram	458	29.76	201.90
Trigram	1,063	33.91	121.54
4-gram	2,036	35.37	107.64
5-gram	3,079	35.70	104.69
6-gram	4,046	35.72	103.95
7-gram	4,938	35.74	103.54

From Table 1, we can see that the improvement of the translation quality is very obvious when unigram is upgraded to bigram and trigram. However, the 6-gram or 7-gram models perform similar to the 5-gram model, but increase the model size a lot. Considering the balance between the model size and the translation performance, we choose the 5-gram model as our baseline. Our main task is to prune the 5-gram language model as much as possible.

To have a thorough investigation, we set the count cutoff threshold from 2 to 70, and see how the relation between the model size and translation quality changes. Figure 1 illustrates the statistics.

Figure 1 demonstrates that the language model size is reduced significantly with the threshold grows, but the translation quality does not decrease significantly ($p \leq 0.5$). In practice, we also test the performance with much stricter threshold (e.g. 100 ~ 500). It turns out that the translation performance drops significantly with the threshold around 100 while the language model size is almost unchanged.

2.3 Language Model Pruning for the BTG-based Translation

The BTG (Bracketing Transduction Grammars) based translation (Wu, 1997; Xiong et al., 2006; Zhang and Zong, 2009) can be viewed as a monolingual parsing process,

[2] SRILM is available at `http://www.speech.sri.com/projects/srilm/`

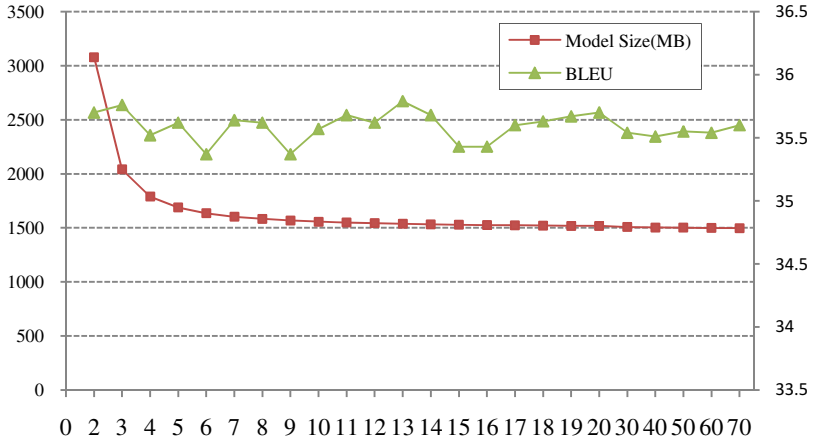

Fig. 1. Performance of the Phrase-based translation with pruned language model. The X-axis represents the threshold for pruning, and the Y-axis denotes the size of the pruned language model size and the translation quality (in BLEU scores).

in which only lexical rules $A \rightarrow (x, y)$ and two binary merging rules $A \rightarrow [A^l, A^r]$ and $A \rightarrow \langle A^l, A^r \rangle$ are allowed.

During decoding, the source language sentence is first divided into phrases (sequence of words), then the lexical translation rule $A \rightarrow (x, y)$ translates each source phrase x into target phrase y and forms a block A. The monotone merging rule $A \rightarrow [A^l, A^r]$ (or the swap merging rule $A \rightarrow \langle A^l, A^r \rangle$) combines the two neighboring blocks into a bigger one until the whole source sentence is covered.

The lexical translation rule $A \rightarrow (x, y)$ plays the same role as the phrasal translation pairs (tuples consisting of a source phrase and its target translation hypothesis) in the conventional phrase-based translation models. The monotone merging rule $A \rightarrow [A^l, A^r]$ combines the two consecutive blocks into a bigger block by concatenating the two partial target translation candidates in order while the swap rule $A \rightarrow \langle A^l, A^r \rangle$ yields the bigger block by swapping the two partial target translation candidates. The probability of the merging rules is estimated using a maximum entropy based algorithm (Xiong et al., 2006).

Similarly, the n-gram language model measures the fluency of the translation outputs. We also apply the same count cutoff pruning technique to figure out the relations between the language model size and the BTG-based translation quality. Table 2 reports the statistics of the baseline n-gram models. The trend is very similar to that of the phrase-based translation, except that the 5-gram model does not outperform significantly over the 4-gram model. In order to keep consistency, we also use the 5-gram model as our baseline for language model pruning.

Table 2. Performance of the BTG-based translation with each n-gram language model

Language Model	Size(MB)	BLEU	Perplexity
Unigram	32	24.29	1155.74
Bigram	458	29.59	201.90
Trigram	1,063	34.92	121.54
4-gram	2,036	36.56	107.64
5-gram	3,079	36.62	104.69
6-gram	4,046	36.68	103.95
7-gram	4,938	36.72	103.54

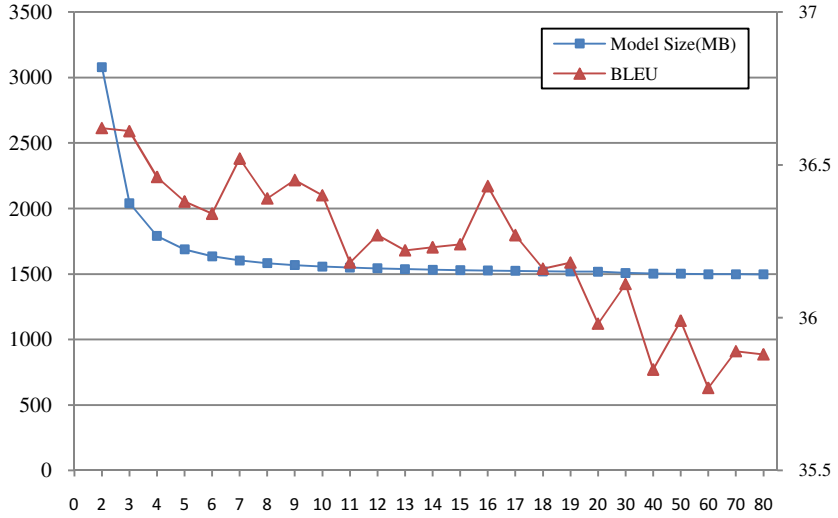

Fig. 2. Performance of BTG-based translation system with pruned language model

Figure 2 gives the statistics of the pruned language model size and the BTG-based translation quality. We can see from the figure that the BTG-based translation system is more sensitive to the count cutoff technique. When the count threshold is bigger than 16, the translation quality decreases rapidly.

2.4 Analysis

The results of pruning imply that the performance of translation can still keep in an acceptable level although the pruned language model is almost only half of the baseline in model size. It sheds new lights to training the language model in practical use. We compare the baseline language model (Table 3) with the pruned one using the threshold value 8, which perform well both in the phrase-based model and BTG-based model, to look into the pruned results in detail.

Table 3. Comparsion between original model and pruned model

N-gram in language model	Baseline	Pruned(threshold value=8)
Unigram	1,715,898	1,715,898
Bigram	16,680,196	16,680,196
Trigram	18,185,555	14,111,461
4-gram	24,364,923	16,234,025
5-gram	25,065,466	1,811,926

Table 3 demonstrates that most part of 5-grams, nearly one third of 4-grams and one fifth of trigrams have been discarded during pruning. Pruning the 5-grams only can also affect the lower order n-grams. This is because during the count cutoff pruning process, the Kneser-Ney smoothing method would modify the lower n-grams (4-grams and trigrams) which the pruned 5-grams are related to.

In general, the performance trends of the two different translation systems are similar with each other and it is consistent with our intuitions. Due to the different translation generation styles, the bottom-up BTG-based translation system is more sensitive to the count threshold compared with the left-to-right phrase-based translation system. We believe it is because that the reordering model in the BTG-based translation system is more powerful than that in the phrase-based system, and the strong reordering model alleviates the dependence on the higher order n-gram language model. However, we will explore the deep reasons in the future work.

According to our analysis, we find another interesting phenomenon: in the pruned language model, many 4-grams and 5-grams share the same trigram which has high relative probability. For example, considering the trigram "*is aimed at*", many 4-grams and 5-grams sharing the same context are shown in Figure 3.

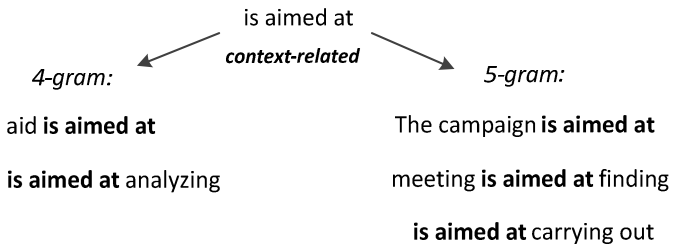

Fig. 3. The trigram "*is aimed at*" is shared in many 4-grams and 5-grams. This figure illustrates some examples.

For better understanding, we further sort all the trigrams by probability in descending order and select the top *k* frequent trigrams to see how many higher order 4-grams and 5-grams share these frequent trigrams in the pruned language model. We report the statistics in Table 4.

Table 4. Influence of the top *k* frequent trigrams in the pruned model. According to statistics in this table, the top 3 million frequent trigrams can cover 41.88% 4-grams and 70.26% 5-grams. The top 5 million frequent trigrams can account for 66.30% 4-grams and 91.11% 5-grams.

k (million)	4-gram	5-gram
1	14.45%	30.97%
2	27.75%	53.27%
3	41.88%	70.26%
4	55.11%	82.98%
5	66.30%	91.11%
6	75.57%	96.24%
7	82.48%	98.51%
8	87.65%	99.39%
9	91.38%	99.69%
10	94.06%	99.86%
>10	>95%	>99.86%

As we can see, the most 4-grams and 5-grams in the pruned language model can be explained with the frequent trigrams. This inspires us to make full use of the frequent trigrams so that we can safely discard all the 4-grams and 5-grams to keep the language model size as small as possible.

3 Frequent Trigram Based Model Pruning

According to our analysis, the frequent trigrams can cover most of the 4-grams and 5-grams. It is natural that we can take full advantage of the frequent trigrams in the language model pruning task.

The idea of capturing the frequent trigram information is not to select the particular n-grams from the language model, but to regard the frequent trigrams as basic language units. Thus, we can treat 4-grams and 5-grams as the new bigram and trigrams related to the original frequent trigrams.

From another point of view, the standard n-gram language model uses the word as the basic unit, and the trigram model can only depict dependence among three words. However, considering the frequent trigrams as basic units, we can capture some relationship in the phrase level. Taking the following English sentence as an example:

The China's policy is aimed at the trade

We find that the phrases '***the China's policy***' and '***is aimed at***' are two frequent trigrams. Therefore, different from the standard word-based trigram language model, we can estimate the probability of this sentence using the new trigram model as below:

$$p(\mathrm{s}) = p(\textbf{\textit{the\_china's\_policy}} \mid \langle \mathrm{s} \rangle)*$$
$$p(\textbf{\textit{is\_aimed\_at}} \mid \textbf{\textit{the\_China's\_policy,}}\ \langle \mathrm{s} \rangle)*$$
$$p(\textit{the} \mid \textbf{\textit{is\_aimed\_at, the\_China's\_policy}})*$$
$$p(\textit{trade} \mid \textbf{\textit{the\_China's\_policy}},\quad \textit{the})*$$
$$p(\langle \mathrm{s} \rangle \mid \textit{the}\ ,\ \textit{trade})$$

By using the above new trigram model, we can fill some gap caused by discarding all the baseline 4-grams and 5-grams. However, it still remains a question how to leverage the baseline frequent trigrams in the translation modeling. In this paper, we propose two simple approaches: one treats the frequent trigrams as concrete basic units, and the other cluster the frequent trigrams and learn a variable (cluster) based language model.

3.1 Frequent Trigrams as Concrete Basic Units

Regarding the baseline frequent trigrams as concrete basic units and learning a new trigram model is the same as training a normal language model. The only difference is that we should replace all the recognized frequent trigrams with the basic unit forms. The training procedure can be processed with four steps as follows:

Step 1: extract all trigrams from the pruned language model which performs well in the translation task.

Step 2: sort the trigrams by probability in descending order, and select the top k trigrams as frequent trigrams.

Step 3: find these trigrams in original language model training corpus and replace them with basic unit forms. For Example, "*is aimed at*" is substitute by "*is_aimed_at*". Then we can obtain the new training corpus in which frequent trigrams are transformed into basic units.

Step 4: train the new trigram language model using the new corpus.

During decoding, we incorporate the frequent trigram model as an another language model feature besides the baseline trigram model, into the log-linear model. The total model score is computed by a log-linear combination of a group of translation features, reordering features and the language models.

$$g(h;t,r,lm,lm_f) = w_t \cdot m_t(h) + w_r \cdot m_r(h) + w_{lm} \cdot m_{lm}(h) + w_{lmf} \cdot m_{lmf}(h) \quad (2)$$

g denotes the total score of the translation hypothesis, t denotes the translation model, r denotes the reordering model, lm denotes the language model and lm_f denotes the new frequent trigram model. w_{lmf} represents the weight of the new frequent trigram model.

When we score a translation hypothesis, we first detect whether there exist frequent trigrams in this hypothesis. When the hypothesis contains the frequent trigrams,

we can get a score from new frequent trigram model. We just set the default score of the frequent trigram model 0 in case there is no frequent trigram in the hypothesis.

3.2 Cluster-Based Model

The frequent trigram model can capture much contextual information. According to the property of the frequent trigram model, it can capture similar or more information compared with the baseline 4-grams or 5-grams.

However, the model size will become unacceptable if we retain much more top trigrams. In order to capture the key information and reduce the size of this model, we propose to use the clustering method to capture the patterns in the frequent trigrams. The brown clustering method has been introduced as a pruning method in Section 2. We also use this algorithm as the clustering technique to cluster frequent trigrams rather than single words in the previous work. Intuitively, compared to the frequent trigram model, our cluster-based trigram model will save much storage and will avoid the data sparseness problem.

In order to perform this approach, we directly run brown clustering algorithm on the training corpus which is used for learning the frequent trigram model as we mentioned in the last section. After clustering, we can obtain the classes for each frequent trigram. Then, the training procedure of the clustering model is processed as follows:

Step 1: build a mapping between frequent trigrams and the classes (number).

Step 2: replace the frequent trigrams in the original corpus with class identifier. For instance, we replace "*is aimed at*" with "*C12*".

Step 3: train the new cluster-based language model.

Like the frequent trigram model, we also integrate the cluster-based language model as a new feature into the log-linear translation model besides the baseline trigram model. During decoding, if we detect the frequent trigram in the translation hypothesis, then we replace it with class identifier and retrieve the probability in the cluster-based model.

In addition, we also discard a lot of uninformative n-grams in our cluster-based model, such as:

(a) unigrams in our cluster-based model, because the unigram information is included in the baseline trigram model.
(b) any n-gram containing no frequent trigrams.

4 Experiments and Discussion

We conduct our experiments on NIST Chinese-to-English translation to evaluate the effectiveness of both the frequent trigram language model and the cluster-based model.

To train our baseline trigram language model, the frequent trigram model and the cluster-based trigram model, we use the same corpus and settings as that used in Section 2.

We use the NIST MT03 evaluation test data as the development set, and the NIST MT04, MT05 as the test sets. We also adopt the case-insensitive BLEU-4 with the shortest length penalty as our evaluation metric. The language model size and translation quality results are shown in Table 5.

Table 5. Experimental results of the frequent trigram languge model (flm) and the cluster-based trigram languge model (clm). 4m means use top 4 milion trigrams as the frequent trigrams. 100c means clustering frequent trigrams into 100 word class.

Model	Size(MB)	MT03(dev)	MT04	MT05
Trigram	1000	35.01	34.92	32.95
trigram+flm(4m)	2000	35.60	35.52	33.47
trigram+clm(4m,100c)	1200	35.57	35.50	33.46

From the results in Table 5, we can see that when we combine the frequent trigram language model with the baseline trigram language model, we can obtain significant improvements.

As to the cluster-based trigram language model, it performs similar to the frequent trigram language model, but it can reduce the model size significantly. Specifically, the cluster-based trigram language model outperforms the baseline trigram model with an improvement more than 0.5 BLEU score, while increasing only 200MB storage. It demonstrates the effectiveness of our proposed cluster-based language model.

We have also tried 4-gram model and 5-gram model as our baselines. However, the improvement obtained by the frequent trigram model and the cluster-based language model is not significant. We believe the reason is that the 4-grams and 5-grams contain most information of the new models. Another reason can be seen from Section 2 that with the n-gram order becomes higher, the improvement becomes smaller and smaller.

5 Conclusions and Future Work

This paper presents an investigation about the relations between the language model size and the translation quality in statistical machine translation. Based on our interesting findings, we further propose to replace the higher order n-grams (4-grams and 5-grams) with a low-order (trigram) cluster-based language model which attempts to make full use of the frequent trigrams. The extensive experiments show that our methods are very effective in both of reducing the language model size and improving the translation quality.

In the future, we will study further the frequent trigrams and find more effective algorithms which can capture the intrinsic structure of the language model.

Acknowledgment. The authors are grateful to *International Science & Technology Cooperation Program of China* (Grant No.2014DFA11350) and *High New Technology Research and Development Program of Xinjiang Uyghur Autonomous Region* (Grant No. 201312103).

References

1. Brown, P., Della Pietra, V., de Souza, P., Lai, J., Mercer, R.: Class-based n-gram models of natural language. Computational Linguistics (18), 467–479 (1990)
2. Goodman, J., Gao, J.: Language model size reduction by pruning and clustering. In: Processings of ICSLP 2000, pp. 110–113 (2000)
3. Jelinek, F., Merialdo, B., Roukos, S., Strauss, M.: Self Organized Language modeling for Speech Recognition. In: Waibel, A., Lee, K.F. (eds.) Reading in Speech Recognition. Morgan Kaufmann (1990)
4. Koehn, P., Hoang, H., Birch, A., Callsion-Burch, C., et al.: Moses: Open Source Toolkit for Statistical Machine Translation. In: Proceedings of ACL, pp. 177–180 (2007)
5. Moore, R.C., Quirk, C.: Less is More: Significance-Based N-gram Selection for Smaller, Better Language Models. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pp. 746–755 (2009)
6. Och, F.J.: GIZA++:Training of statistical translation models (2000)
7. `http://www-i6.informatik.rwth-aachen.de/~och/software/GIZA++.html`
8. Seymore, K., Rosenfeld, R.: Scalable Trigram Backoff Language Models. In: Proceedings of ICSLP 1996, pp. 232–235 (1996)
9. Stolcke, A.: Entropy-based pruning of backoff language model. In: Proceedings of the DARPA News Transcription and Understanding Workshop 1998, pp. 270–274 (1998)
10. Wu, D.: Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistic 23(3), 377–403 (1997)
11. Xiong, D., Liu, Q., Lin, S.: Maximum entropy based phrase reordering model for statistical machine translation. In: Proceedings of COLING-ACL 2006, pp. 521–528 (2006)
12. Zhang, J., Zong, C.: A Framework for Effectively Integrating Hard and Soft Syntactic Rules into Phrase-Based Translation. In: Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation (PACLIC 23), Hong Kong, pp. 579–588 (2009)