Natural Language Engineering **21** (2): 201–226. © Cambridge University Press 2013
doi:10.1017/S1351324913000168

Backward and trigger-based language models for statistical machine translation

DEYI XIONG and MIN ZHANG
School of Computer Science and Technology,
Soochow University, Suzhou 215006, China
email: dyxiong@suda.edu.cn,minzhang@suda.edu.cn

(*Received 9 March 2012; revised 19 June 2013; accepted 26 June 2013;*
first published online 24 July 2013)

Abstract

The language model is one of the most important knowledge sources for statistical machine translation. In this article, we present two extensions to standard *n*-gram language models in statistical machine translation: a backward language model that augments the conventional forward language model, and a mutual information trigger model which captures long-distance dependencies that go beyond the scope of standard *n*-gram language models. We introduce algorithms to integrate the two proposed models into two kinds of state-of-the-art phrase-based decoders. Our experimental results on Chinese/Spanish/Vietnamese-to-English show that both models are able to significantly improve translation quality in terms of BLEU and METEOR over a competitive baseline.

1 Introduction

In statistical machine translation (SMT), the widely used standard *n*-gram language model assigns probabilities to hypotheses in the target language conditioning on a context history of the preceding $n-1$ words. With a belief that a language model that embraces a larger context provides better prediction ability, we introduce a backward language model (BLM) and a mutual information (MI) trigger model into SMT. Using these two models, we attempt to enhance the ability of conventional *n*-gram language models in capturing richer contexts and long-distance dependencies.

Conventional *n*-gram language models look at the preceding $n-1$ words when calculating the probability of the current word. We henceforth call the previous $n-1$ words plus the current word as *forward n-grams* and a language model built on forward *n*-grams as the forward *n*-gram language model. Similarly, *backward n-grams* refer to the succeeding $n-1$ words plus the current word. We train a backward *n*-gram language model on backward *n*-grams and integrate the forward and backward language models together into our decoder. In doing so, we are able to capture both the preceding and succeeding contexts of the current word.

Different from the backward *n*-gram language model, the MI trigger model still looks at preceding context, which however goes beyond the scope of forward *n*-grams. If the current word is indexed as w_i, the farthest word that the forward *n*-gram

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

includes is w_{i-n+1}. However, the MI triggers are capable of detecting dependencies between w_i and words from w_1 to w_{i-n}. Using these triggers ($\{w_k \rightarrow w_i\}, 1 \leq k \leq i-n$), we can capture long-distance dependencies that are outside the scope of forward n-grams.

We integrate the two new language models into a state-of-the-art bracketing transduction grammar (Wu 1997) (BTG)-based system (Xiong, Liu and Lin 2006). The integration algorithms for the backward language model and MI trigger model are described in details. We also introduce how the two models are integrated into the standard phrase-based system (Koehn, Och and Marcu 2003). We conduct a series of experiments with BTG-based systems to validate the effectiveness of the two models.

This article significantly extends our previous work (Xiong, Zhang and Li 2011) in four major aspects.

- Firstly, we thoroughly compare the single backward language model with (1) the single forward language model (FLM) and (2) the combination of the forward and backward language models by integrating them into the decoder at the same time.
- Secondly, we investigate different trigger pair selection criteria in order to maximize the performance of the MI trigger model.
- Thirdly, we explore distance-dependent and syntactically informed triggers in order to investigate whether such triggers are able to obtain further improvements.
- Finally, we validate the robustness of the proposed models on several language pairs, including Chinese-to-English, Spanish-to-English and Vietnamese-to-English.

The remaining of this article is organized as follows. Section 2 briefly introduces related work and shows how our models differ from previous work. Section 3 presents the model and decoder of our BTG-based baseline system. Sections 4 and 5 elaborate the backward language model and the MI trigger model respectively and describe the training procedures for the two models. Section 6 discusses the algorithms that integrate the backward language model and MI trigger model into the BTG-based system. Section 7 introduces how we integrate the two models into the standard phrase-based SMT system. Section 8 empirically evaluates the effectiveness of these two models using BTG-based SMT systems. In the end, we conclude in Section 9.

2 Related work

Previous work devoted to improving language models in SMT mostly focuses on two categories[1]: large language models (Zhang, Hildebrand and Vogel 2006; Brants *et al.* 2007; Emami , Papineni and Sorensen 2007; Talbot and Osborne 2007) and syntax-based language models (Charniak, Knight and Yamada 2003; Post and Gildea 2008;

[1] Language model adaptation is not very related to our work so we ignore it.

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

Shen, Xu and Weischedel 2008). The former language models are built on large-scale monolingual corpora (LDC news data or news data collected from web pages) that consist of billions or even trillions of words. The latter language models explore linguistic knowledge (e.g., syntactical structures) of monolingual data to capture long-distance dependencies by resorting to linguistic parsers. Our philosophy is fundamentally different from them in that we build contextually informed language models by using backward *n*-grams and MI triggers. Therefore, we focus on previous work that explores these two techniques (backward *n*-grams and MI triggers).

Since the context 'history' in the backward language model is actually the future words to be generated, the backward language model is normally used in a post-processing where all words have already been generated or in a scenario where sentences are reversed. Duchateau, Demuynck and Wambacq (2002) use the backward language model score as a confidence measure to detect wrongly recognized words in speech recognition. Finch and Sumita (2009) use the backward language model in their reverse translation decoder where source sentences are reversed. Our backward language model is different from theirs in that we access the backward language model during decoding (rather than after decoding) where source sentences are not reversed. In this way, we can concurrently integrate both the forward and backward language models into the decoder in a tight manner, rather than creating two separate decoding instances with the two language models, respectively (Finch and Sumita 2009).

Rosenfeld, Carbonell and Rudnicky (1994) introduce trigger pairs into a maximum entropy-based language model as features. The trigger pairs are selected according to their MI. Zhou (2004) also proposes an enhanced language model (MI-Ngram) which consists of a standard forward *n*-gram language model and an MI trigger model. The latter model measures the MI of distance-dependent trigger pairs. Our MI trigger model is mostly inspired by the work of these two articles, especially by Zhou's MI-Ngram model (Zhou 2004). The difference is that our model is distance independent and, of course, we are interested in an SMT problem rather than a speech recognition one.

Raybaud *et al.* (2009) use MI triggers in their confidence measures to assess the quality of translation results after decoding. Our method is different from theirs not only in that we use triggers during decoding rather than after decoding but also in the MI calculation and trigger pair selection. Mauser, Hasan and Ney (2009) propose bilingual triggers where two source words trigger one target word to improve lexical choice of target words. Our experiments (Section 8.3.3) show that our monolingual triggers can also help in the selection of target words.

3 Baseline system: BTG-based SMT

In this section, we briefly introduce our baseline BTG-based SMT system that adapts BTGs to phrasal translation. The introduction includes two parts: the BTG-based SMT model and the Cocke–Younger–Kasami (CKY)-style decoder. Both of them are helpful for readers to understand the language model integration algorithms described in Section 6.

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

3.1 Model

The log-linear model of the baseline BTG-based SMT system is identical to the one defined in Xiong *et al.* (2006), which is formulated as follows.

$$M(\mathscr{D}) = M_T(r^l_{1..n_l}) \cdot M_R(r^m_{1..n_m})^{\lambda_R} \cdot P_f(e(\mathscr{D}))^{\lambda_f} \cdot \exp(|e(\mathscr{D})|)^{\lambda_w} \quad (1)$$

where

- $\mathscr{D}$ denotes a BTG derivation that is defined as a sequence of BTG rules.

$$\mathscr{D} =< r^l_{1..n_l}, r^m_{1..n_m} >$$

- $r^l_{1..n_l}$ are BTG lexical rules which translate source phrases f into target phrases e as follows.

$$r^l : X \rightarrow e/f$$

- $r^m_{1..n_m}$ are BTG merging rules that combine two neighboring blocks X_1 and X_2 into a larger block X in a straight ([ ]) or inverted ($\langle\ \rangle$)order.

$$r^m : X \rightarrow [X_1, X_2] \text{ or } \langle X_1, X_2 \rangle$$

- The translation model $M_T(r^l_{1..n_l})$ consists of phrase and lexical translation probabilities that are widely used in phrase-based SMT (Koehn *et al.* 2003) for the BTG lexical rules $r^l_{1..n_l}$.
- The reordering model $M_R(r^m_{1..n_m})$ predicts the merging order (straight or inverted) with discriminative contextual features (Xiong *et al.* 2006) for the BTG merging rules $r^m_{1..n_m}$.
- P_f is the standard forward n-gram language model.
- $\exp(|e(\mathscr{D})|)$ represents the word penalty model on the target yield $e(\mathscr{D})$.
- λs are weights of sub-models mentioned above (i.e., $\{M_T, M_R, P_f, \exp(||)\}$, which are tuned by minimum error rate training (MERT) (Och 2003).

3.2 Decoder

We use a CKY-style decoder to find the best translation for each source sentence according to the model $M(\mathscr{D})$ defined in (1). In particular, given a source sentence $f_1 \ldots f_J$, the CKY-style decoder employs BTG rules to generate derivations for each segment spanning from f_i to f_j. When the decoding procedure completes, the best derivation $\mathscr{D}^*$ that covers the whole input sentence will be found. The final translation e^* is produced from the best derivation as follows.

$$\mathscr{D}^* = \underset{f(\mathscr{D})=f_1 \ldots f_J}{\operatorname{argmax}} M(\mathscr{D})$$
$$e^* = e(\mathscr{D}^*) \quad (2)$$

where $f(\mathscr{D})$ and $e(\mathscr{D})$ are the source and target yields of $\mathscr{D}$, respectively.

Following Chiang (2007), we describe the CKY-style decoder for BTG-based SMT in a deductive system. The decoding algorithm is shown in Figure 1. $[X, i, j, \mathscr{S}(X)] : \mathscr{W}$ denotes a weighted item X spanning from f_i to f_j with a weight $\mathscr{W}$ which is calculated according to (1). $\mathscr{S}(X)$ is the n-gram language model state of the item.

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

$$\frac{X \rightarrow e/f}{[X, i, j, \mathcal{S}(X)] : \mathcal{W}} \tag{3}$$

$$\frac{X \rightarrow [X_1, X_2] \;\; [X_1, i, k, \mathcal{S}(X_1)] : \mathcal{W}_1 \;\; [X_2, k+1, j, \mathcal{S}(X_2)] : \mathcal{W}_2}{[X, i, j, \mathcal{S}(X)] : \mathcal{W}_1 \mathcal{W}_2 \Delta(X \rightarrow [X_1, X_2])} \tag{4}$$

$$\frac{X \rightarrow \langle X_1, X_2 \rangle \;\; [X_1, i, k, \mathcal{S}(X_1)] : \mathcal{W}_1 \;\; [X_2, k+1, j, \mathcal{S}(X_2)] : \mathcal{W}_2}{[X, i, j, \mathcal{S}(X)] : \mathcal{W}_1 \mathcal{W}_2 \Delta(X \rightarrow \langle X_1, X_2 \rangle)} \tag{5}$$

Fig. 1. The CKY-style decoding algorithm for BTG-based SMT.

Procedure CKYDecode
for all lexical rules $X \rightarrow e/f$ **do**
  add item $[X, i, j, \mathcal{S}(X)] : \mathcal{W}$ to $chart[X, i, j]$
end for
for span = 1 to J **do**
  for i = 1 to J-span+1 **do**
    j = i+span-1
    for k = i to j -1 **do**
      for all items $[X, i, k, \mathcal{S}(X_1)] : \mathcal{W}_1$ and $[X, k+1, j, \mathcal{S}(X_2)] : \mathcal{W}_2$ **do**
        add $[X, i, j, \mathcal{S}(X)] : \mathcal{W}_1 \mathcal{W}_2 \Delta(X \rightarrow [X_1, X_2])$ to $chart[X, i, j]$
        add $[X, i, j, \mathcal{S}(X)] : \mathcal{W}_1 \mathcal{W}_2 \Delta(X \rightarrow \langle X_1, X_2 \rangle)$ to $chart[X, i, j]$
      end for
    end for
  end for
end for

Fig. 2. The CKY-style decoding procedure for BTG-based SMT.

Normally, we define the leftmost/rightmost $n-1$ words in target translations as the language model state for CKY-style decoding. In the algorithm, (3) generates an item using the lexical rule $X \rightarrow e/f$ to translate the source phrase f spanning from i to j to the target phrase e. Equation (4) merges two neighboring items into a larger item in a straight order and (5) in an inverted order. The weights of new items are calculated in a dynamic programming manner by reusing weights $\mathcal{W}_1$ and $\mathcal{W}_2$ of their sub-items. We only need to compute the incremental part Δ that includes the language model increment which will be described in Section 6 and the reordering model score for the BTG merging rules in a straight or inverted order (Xiong *et al.* 2006).

The actual CKY-style decoding procedure is given by the pseudo code in Figure 2. All items spanning from i to j are stored in a chart cell $chart[X, i, j]$. When we add new items to a chart cell, we recombine items with the same language model state by discarding those with lower weights in order to trim the search space.

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

4 Backward language model

4.1 Model

Given a sequence of words $w_1^m = (w_1 \dots w_m)$, a standard forward n-gram language model assigns a probability $P_f(w_1^m)$ to w_1^m as follows.

$$P_f\left(w_1^m\right) = \prod_{i=1}^{m} P\left(w_i | w_1^{i-1}\right) \approx \prod_{i=1}^{m} P\left(w_i | w_{i-n+1}^{i-1}\right) \tag{6}$$

where the approximation is based on the nth order Markov assumption. In other words, when we predict the current word w_i, we only consider the preceding $n-1$ words $w_{i-n+1} \dots w_{i-1}$ instead of the whole context history $w_1 \dots w_{i-1}$.

Different from the forward n-gram language model, the backward n-gram language model assigns a probability $P_b(w_1^m)$ to w_1^m by looking at the succeeding context according to

$$P_b\left(w_1^m\right) = \prod_{i=1}^{m} P\left(w_i | w_{i+1}^{m}\right) \approx \prod_{i=1}^{m} P\left(w_i | w_{i+1}^{i+n-1}\right) \tag{7}$$

4.2 Training

For the convenience of training, we invert the order in each sentence in the training data, i.e., from the original order $(w_1 \dots w_m)$ to the reverse order $(w_m \dots w_1)$. In this way, we can use the same toolkit that we use to train a forward n-gram language model to train a backward n-gram language model without any other changes. To be consistent with training, we also need to reverse the order of translation hypotheses when we access the trained backward language model.[2] Note that the Markov context history of (7) is $w_{i+n-1} \dots w_{i+1}$ instead of $w_{i+1} \dots w_{i+n-1}$ after we invert the order. The words are the same but the order is completely reversed.

5 MI trigger model

It is well known that long-distance dependencies between words are very important for statistical language modeling. However, n-gram language models can only capture short-distance dependencies within an n-word window. In order to model long-distance dependencies, previous works such as Rosenfeld *et al.* (1994) and Zhou (2004) exploit trigger pairs in speech recognition. A trigger pair is defined as an ordered pair (x, y), where word x occurs in the preceding context of word y. It can also be denoted in a more visual manner as $x \rightarrow y$ with x being the trigger and y the triggered word. In this section, we introduce our MI trigger model, which uses trigger pairs to capture long-distance dependencies for SMT.

[2] This is different from the reverse decoding in Finch and Sumita (2009), where source sentences are reversed in the order.

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

5.1 Model

We use pointwise mutual information (PMI) (Church and Hanks 1990) to measure the strength of the association between x and y in a trigger pair, which is defined as follows

$$PMI(x, y) = \log\left(\frac{P(x, y)}{P(x)P(y)}\right) \tag{8}$$

where $P(x, y)$ is the joint probability of (x, y), and $P(x)$ and $P(y)$ are marginal probabilities that can be derived from $P(x, y)$.

Zhou (2004) proposes a new language model enhanced with MI trigger pairs. In his model, the probability of a given sentence w_1^m is approximated as

$$\begin{aligned} P\left(w_1^m\right) \approx & \left(\prod_{i=1}^{m} P\left(w_i | w_{i-n+1}^{i-1}\right)\right) \\ & \times \prod_{i=n+1}^{m} \prod_{k=1}^{i-n} \exp(PMI(w_k, w_i, i-k-1)) \end{aligned} \tag{9}$$

There are two components in his model. The first component is still the standard n-gram language model. The second one is the MI trigger model which multiplies all exponential PMI values for trigger pairs where the current word is the triggered word and all preceding words outside the n-gram window of the current word are triggers. Note that his MI trigger model is distance dependent since trigger pairs (w_k, w_i) are sensitive to their distance $i-k-1$ (zero distance for adjacent words). Therefore, the distance between word x and word y should be taken into account when calculating their PMI.

In this article, we adopt a distance-independent MI trigger model as follows

$$MI\left(w_1^m\right) = \prod_{i=n+1}^{m} \prod_{k=1}^{i-n} \exp(PMI(w_k, w_i)) \tag{10}$$

We integrate the MI trigger model into the log-linear model of machine translation as an additional knowledge source which complements the standard n-gram language model in capturing long-distance dependencies. Using MERT (Och 2003), we are even able to tune the weight of the MI trigger model against the weight of the standard n-gram language model while Zhou (2004) sets equal weights for both models.[3]

In Section 8.3.2, we empirically compare our distance-independent MI trigger model with Zhou's distance-dependent model.

5.2 Training

We can use the maximum likelihood estimation method to calculate PMI for each trigger pair by taking counts from the training data. Let $C(x, y)$ be the number of

[3] We show the relative weights of these two language models in our experiments in Table 11.

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

times that x and y co-occur in the same sentence in the training data. The joint probability of (x, y) is calculated as

$$P(x, y) = \frac{C(x, y)}{\sum_x \sum_y C(x, y)} \tag{11}$$

The marginal probabilities of x and y can be deduced from the joint probability as follows

$$P(x) = \sum_y P(x, y) \tag{12}$$

$$P(y) = \sum_x P(x, y) \tag{13}$$

Since the number of distinct trigger pairs is $\mathcal{O}(|T|^2)$, where T is the target language vocabulary, the question is how to select valuable trigger pairs. We select trigger pairs according to the following three steps.

(1) The distance between x and y must not be less than $n - 1$. Suppose, we use a 5-gram language model and $y = w_i$, then $x \in \{w_1 \dots w_{i-5}\}$.
(2) $C(x, y) > c$. In all our experiments, we set $c = 10$.
(3) Finally, we only keep trigger pairs whose PMI value is larger than 0. Trigger pairs whose PMI value is less than 0 often contain stop words, such as 'the', 'a'. These stop words have very large marginal probabilities due to their high frequencies. In Section 8.3.1, we empirically investigate whether we should use PMI-negative trigger pairs or trigger pairs with stop words.

6 Integrating the two models into BTG-based SMT

6.1 Backward language model integration

In this section, we present an algorithm to integrate the backward n-gram language model into the CKY-style decoder of BTG-based SMT that is described in Section 3.2. Wu (1996) introduces a dynamic programming algorithm to integrate a forward bigram language model with inversion transduction grammar. His algorithm is then adapted and extended for integrating forward n-gram language models into synchronous context free grammars (CFGs) by Chiang (2007). Our algorithm is different from theirs in two major aspects as follows.

(1) The string input to the algorithms is in a reverse order.[4]
(2) We adopt a different way to calculate language model probabilities for partial hypotheses so that we can utilize incomplete n-grams.

Before we introduce the integration algorithm, we define three functions $\mathscr{P}$, $\mathscr{L}$, and $\mathscr{R}$ on strings (in a reverse order) over the target language terminal alphabet T.

[4] Note that we do not reverse source sentences. We just reverse target translations when we calculate the backward language model scores of them.

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

The function $\mathscr{P}$ is defined as follows.

$$\mathscr{P}(w_k \dots w_1) = \underbrace{P(w_k) \dots P(w_{k-n+2}|w_k \dots w_{k-n+3})}_{a} \times \underbrace{\prod_{1 \le i \le k-n+1} P(w_i|w_{i+n-1} \dots w_{i+1})}_{b} \tag{14}$$

This function consists of two parts:

- The first part (a) calculates incomplete n-gram language model probabilities for word w_k to w_{k-n+2}. That means, we calculate the unigram probability for w_k ($P(w_k)$), bigram probability for w_{k-1} ($P(w_{k-1}|w_k)$) and so on until we take $n-1$-gram probability for w_{k-n+2} ($P(w_{k-n+2}|w_k \dots w_{k-n+3})$). This resembles the way in which the forward language model probability in the future cost is computed in the standard phrase-based SMT (Koehn *et al.* 2003).
- The second part (b) calculates complete n-gram backward language model probabilities for word w_{k-n+1} to w_1.

The function is different from Chiang's p function in that his function p only calculates language model probabilities for the complete n-grams. This makes two items not directly comparable. In order to address this issue, Chiang also calculates incomplete n-gram language model probabilities using a heuristic function. However, this heuristic function for incomplete n-grams is added to the score of an item only for pruning. In our integration algorithm, language model probabilities for incomplete n-grams are intrinsic to item scores.

Since we calculate backward language model probabilities in a left-to-right decoding process, the succeeding context for the current word is either yet to be generated or incomplete in terms of n-grams. The $\mathscr{P}$ function enables us to utilize incomplete succeeding contexts to approximately predict words. Once the succeeding contexts are complete, we can quickly update language model probabilities in an efficient way in our algorithms.

The other two functions $\mathscr{L}$ and $\mathscr{R}$ are defined as follows

$$\mathscr{L}(w_k \dots w_1) = \begin{cases} w_k \dots w_{k-n+2}, & \text{if } k \ge n \\ w_k \dots w_1, & \text{otherwise} \end{cases} \tag{15}$$

$$\mathscr{R}(w_k \dots w_1) = \begin{cases} w_{n-1} \dots w_1, & \text{if } k \ge n \\ w_k \dots w_1, & \text{otherwise} \end{cases} \tag{16}$$

The $\mathscr{L}$ and $\mathscr{R}$ functions return the leftmost and rightmost $n-1$ words from a string in a reverse order, respectively.

The algorithm is shown in Figure 3. The item $[X, i, j; l|r]$ indicates that a BTG node X has been constructed spanning from i to j on the source side with the leftmost|rightmost $n-1$ words $l|r$ on the target side (the language model state). As mentioned before, all target strings assessed by the defined functions ($\mathscr{P}$, $\mathscr{L}$, and $\mathscr{R}$) are in an inverted order (denoted by $\overline{e}$). In order to highlight how we integrate the backward language model into BTG-based SMT, we only display the backward language model probability for each item, ignoring all other scores

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

Table 1. *Values of $\mathscr{P}$, $\mathscr{L}$, and $\mathscr{R}$ in a 3-gram example*

Function	Value
e_1	$a_1a_2a_3$
e_2	$b_1b_2b_3$
$\mathscr{R}(\overline{e_2})$	b_2b_1
$\mathscr{L}(\overline{e_1})$	a_3a_2
$\mathscr{P}(\mathscr{R}(\overline{e_2}))$	$P(b_2)P(b_1\|b_2)$
$\mathscr{P}(\mathscr{L}(\overline{e_1}))$	$P(a_3)P(a_2\|a_3)$
$\mathscr{P}(\overline{e_1})$	$P(a_3)P(a_2\|a_3)P(a_1\|a_3a_2)$
$\mathscr{P}(\overline{e_2})$	$P(b_3)P(b_2\|b_3)P(b_1\|b_3b_2)$
$\mathscr{P}(\mathscr{R}(\overline{e_2})\mathscr{L}(\overline{e_1}))$	$P(b_2)P(b_1\|b_2)P(a_3\|b_2b_1)P(a_2\|b_1a_3)$
$\mathscr{P}(\overline{e_1e_2})$	$P(b_3)P(b_2\|b_3)P(b_1\|b_3b_2)P(a_3\|b_2b_1)P(a_2\|b_1a_3)P(a_1\|a_3a_2)$

such as phrase translation probabilities. Equation (19) in Figure 3 shows how we calculate the backward language model probability for an axiom which applies a BTG lexical rule to translate a source phrase f into a target phrase e. Equations (20) and (21) show how we update the backward language model probabilities for two merging rules which combine two neighboring blocks in a straight and inverted order respectively. The fundamental theories behind this update are

$$\mathscr{P}(\overline{e_1e_2}) = \mathscr{P}(\overline{e_1})\mathscr{P}(\overline{e_2})\frac{\mathscr{P}(\mathscr{R}(\overline{e_2})\mathscr{L}(\overline{e_1}))}{\mathscr{P}(\mathscr{R}(\overline{e_2}))\mathscr{P}(\mathscr{L}(\overline{e_1}))} \tag{17}$$

$$\mathscr{P}(\overline{e_2e_1}) = \mathscr{P}(\overline{e_1})\mathscr{P}(\overline{e_2})\frac{\mathscr{P}(\mathscr{R}(\overline{e_1})\mathscr{L}(\overline{e_2}))}{\mathscr{P}(\mathscr{R}(\overline{e_1}))\mathscr{P}(\mathscr{L}(\overline{e_2}))} \tag{18}$$

Whenever two strings e_1 and e_2 are concatenated in a straight or inverted order, we can reuse their $\mathscr{P}$ values ($\mathscr{P}(\overline{e_1})$ and $\mathscr{P}(\overline{e_2})$) in terms of dynamic programming. Only the probabilities of boundary words (e.g., $\mathscr{R}(\overline{e_2})\mathscr{L}(\overline{e_1})$ in (17)) need to be recalculated since they have complete n-grams after the concatenation. Table 1 shows values of $\mathscr{P}$, $\mathscr{L}$, and $\mathscr{R}$ in a 3-gram example which helpsto verify (17). These

$$\frac{X \rightarrow e/f}{[X,i,j;\mathcal{L}(\overline{e})|\mathcal{R}(\overline{e})] : \mathcal{P}(\overline{e})} \tag{19}$$

$$\frac{X \rightarrow [X_1,X_2] \quad [X_1,i,k;\mathcal{L}(\overline{e_1})|\mathcal{R}(\overline{e_1})] : \mathcal{P}(\overline{e_1}) \quad [X_2,k+1,j;\mathcal{L}(\overline{e_2})|\mathcal{R}(\overline{e_2})] : \mathcal{P}(\overline{e_2})}{[X,i,j;\mathcal{L}(\overline{e_1e_2})|\mathcal{R}(\overline{e_1e_2})] : \mathcal{P}(\overline{e_1})\mathcal{P}(\overline{e_2})\frac{\mathcal{P}(\mathcal{R}(\overline{e_2})\mathcal{L}(\overline{e_1}))}{\mathcal{P}(\mathcal{R}(\overline{e_2}))\mathcal{P}(\mathcal{L}(\overline{e_1}))}} \tag{20}$$

$$\frac{X \rightarrow \langle X_1,X_2\rangle \quad [X_1,i,k;\mathcal{L}(\overline{e_1})|\mathcal{R}(\overline{e_1})] : \mathcal{P}(\overline{e_1}) \quad [X_2,k+1,j;\mathcal{L}(\overline{e_2})|\mathcal{R}(\overline{e_2})] : \mathcal{P}(\overline{e_2})}{[X,i,j;\mathcal{L}(\overline{e_2e_1})|\mathcal{R}(\overline{e_2e_1})] : \mathcal{P}(\overline{e_1})\mathcal{P}(\overline{e_2})\frac{\mathcal{P}(\mathcal{R}(\overline{e_1})\mathcal{L}(\overline{e_2}))}{\mathcal{P}(\mathcal{R}(\overline{e_1}))\mathcal{P}(\mathcal{L}(\overline{e_2}))}} \tag{21}$$

Fig. 3. Integrating the backward language model into a BTG-based CKY-style decoder.

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

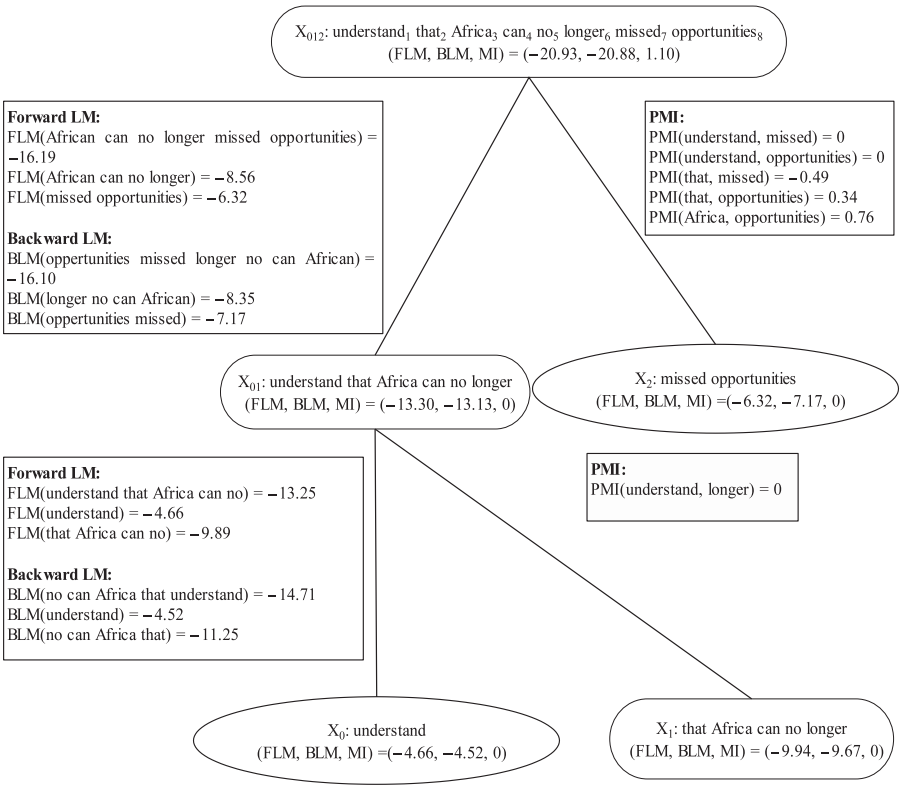


Fig. 4. A BTG tree fragment showing how the values of the forward (FLM), backward (BLM) and mutual information trigger (MI) language model are computed. The rectangles display the FLM/BLM costs (log (base-10) probabilities) of target strings as well as the PMI values of trigger pairs that are used to calculate the FLM/BLM/MI values of items X_{01} (the combination of items X_0 and X_1) and X_{012} (the combination of items X_{01} and X_2).

two equations guarantee that our algorithm can correctly compute the backward language model probability of a sentence stepwise in a dynamic programming framework.[5]

Figure 4 displays a real-world example to show how we update the score of a 5-gram backward language model when two items are merged. Let us take the item X_{01} for instance. This item is the combination of items X_0 (with target string e_1 = 'understand') and X_1 (with target string e_2 = 'that Africa can no longer') in a straight order. The backward language model costs (i.e., $\log\mathscr{P}(\overline{e_1})$ and $\log\mathscr{P}(\overline{e_2})$) of items X_0 and X_1 are −4.52 and −9.67, respectively. The leftmost four words of $\overline{e_1}$ (i.e., $\mathscr{L}(\overline{e_1})$) comprise only the single word 'understand' while the rightmost four words of $\overline{e_2}$ (i.e., $\mathscr{R}(\overline{e_2})$) are 'no can Africa that'. According to (17), the backward language model cost of the item X_{01} should be (−4.52) + (−9.67) + (−14.71) − (−4.52) − (−11.25)

[5] The start-of-sentence symbol $\langle s\rangle$ and end-of-sentence symbol $\langle/s\rangle$ can be easily added to update the final language model probability when a translation hypothesis covering the whole source sentence is completed.

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

$= -13.13$. Similarly, we can easily calculate the backward language model score of item X_{012} when we merge items X_{01} and X_2. Note that we can store the calculated backward language model scores of the leftmost/rightmost words of each item in practice to save time.

The theoretical time complexity of this algorithm is $\mathcal{O}(J^3|T|^{4(n-1)})$ because in the update parts in (17) and (18) both the numerator and denominator have up to $2(n-1)$ terminal symbols. J is the length of source sentence. This is the same as the time complexity of Chiang's language model integration (Chiang 2007).

This algorithm can also be easily adapted to integrate the forward n-gram language model by redefining the three functions $\mathscr{P}$, $\mathscr{L}$, and $\mathscr{R}$ on strings that are not reversed (i.e., in their original order). Due to the space limit, we do not introduce it here. Interesting readers can find how we calculate the forward language model in Figure 4.

6.2 MI trigger model integration

We integrate the MI trigger model into the BTG-based SMT system still in a dynamic programming fashion. In particular, we calculate MI trigger model scores for two strings e_1 and e_2 that are concatenated in a straight (e_1e_2) or inverted (e_2e_1) order as follows.

$$MI(e_1e_2) = MI(e_1)MI(e_2)MI(e_1 \mapsto e_2) \tag{22}$$

$$MI(e_2e_1) = MI(e_1)MI(e_2)MI(e_2 \mapsto e_1) \tag{23}$$

where $MI(e_1 \mapsto e_2)$ represents the MI trigger model score that is calculated by enumerating all trigger pairs in which a word in e_1 triggers a word in e_2. It is defined as follows.

$$MI(e_1 \rightarrow e_2) = \prod_{w_i \in e_2} \prod_{\substack{w_k \in e_1 \\ i-k \geq n}} \exp(PMI(w_k, w_i)) \tag{24}$$

Similarly, $MI(e_2 \mapsto e_1)$ denotes the MI trigger model score for trigger pairs in which a word in e_2 triggers a word in e_1, which is calculated as follows.

$$MI(e_2 \rightarrow e_1) = \prod_{w_i \in e_1} \prod_{\substack{w_k \in e_2 \\ i-k \geq n}} \exp(PMI(w_k, w_i)) \tag{25}$$

The integration algorithm is shown in Figure 5. The problem here is that the state of the MI trigger model involves all words in a string. It is different from the state of the n-gram language model as the latter only considers the preceding or succeeding $n-1$ words. If we define the MI trigger model state as the whole string involved, it results in an intractable integration algorithm as we can not resort to hypotheses recombination for search space pruning. In order to make the integration algorithm tractable, we still use the outermost $n-1$ words to define the MI trigger model state. Although this will lead to search errors, we can benefit from the MI trigger model for long-distance dependencies in a tractable way.

Let us revisit Figure 4 in order to get a clear picture of how we calculate the MI trigger language model. This time we take the item X_{012} (the combination of

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

$$\frac{X \rightarrow e/f}{[X, i, j, \mathcal{L}(e)|\mathcal{R}(e)] : MI(e)} \tag{26}$$

$$\frac{X \rightarrow [X_1, X_2] \;\; [X_1, i, k, \mathcal{L}(e_1)|\mathcal{R}(e_1)] : MI(e_1) \;\; [X_2, k+1, j, \mathcal{L}(e_2)|\mathcal{R}(e_2)] : MI(e_2)}{[X, i, j, \mathcal{L}(e)|\mathcal{R}(e)] : MI(e_1)MI(e_2)MI(e_1 \mapsto e_2)} \tag{27}$$

$$\frac{X \rightarrow \langle X_1, X_2 \rangle \;\; [X_1, i, k, \mathcal{L}(e_1)|\mathcal{R}(e_1)] : MI(e_1) \;\; [X_2, k+1, j, \mathcal{L}(e_2)|\mathcal{R}(e_2)] : MI(e_2)}{[X, i, j, \mathcal{L}(e)|\mathcal{R}(e)] : MI(e_1)MI(e_2)MI(e_2 \mapsto e_1)} \tag{28}$$

Fig. 5. Integrating the MI trigger language model into a BTG-based CKY-style decoder.

$$\frac{[\mathcal{V}; \mathcal{L}(\overline{e_1})] : \mathcal{P}(\overline{e_1}) \quad e_2/f : \mathcal{P}(\overline{e_2})}{[\mathcal{V}'; \mathcal{L}(\overline{e_1 e_2})] : \mathcal{P}(\overline{e_1})\mathcal{P}(\overline{e_2}) \frac{\mathcal{P}(\mathcal{R}(\overline{e_2})\mathcal{L}(\overline{e_1}))}{\mathcal{P}(\mathcal{R}(\overline{e_2}))\mathcal{P}(\mathcal{L}(\overline{e_1}))}} \tag{29}$$

Fig. 6. Integrating the backward language model into a standard phrase-based decoder.

items X_{01} and X_2) as an example. The MI trigger language model score of item X_2 (log value of (10)) is 0 as there are no two words whose distance is equal or larger than four. For item X_{01}, although the distance of the two words (understand, longer) is equal to four, they are not selected as a trigger pair as the two words co-occur rarely. Therefore, the default MI value for these two words is set 0. When we combine items X_{01} and X_2 into item X_{012}, the MI trigger language model score of item X_{012} will be deduced from MI(X_{01}), MI(X_2) and all trigger pairs (x, y) where x and y are in X_{01} and X_2 respectively and their distance is not less than four. The qualified trigger pairs therefore include (understand, missed), (understand, opportunities), (that, missed), (that, opportunities) and (Africa, opportunities). Their PMI values are shown in the PMI rectangle in Figure 4. As we only use trigger pairs whose PMI value is larger than 0, the final MI trigger language model score of item X_{012} is calculated as MI(X_{01}) + MI(X_2) + PMI(that, opportunities) + PMI(Africa, opportunities) = 1.10 according to (24).

7 Integrating the two models into phrase-based SMT

In this section, we adapt the integration algorithms designed for BTG-based SMT to phrase-based SMT (Koehn *et al.* 2003). Figure 6 shows the algorithm for integrating the backward language model into phrase-based SMT. The three functions $\mathscr{P}$, $\mathscr{L}$, and $\mathscr{R}$ are exactly the same as the ones defined in the last section. $\mathscr{V}$ denotes a coverage vector which records source words translated so far. Equation (29) shows how we update the backward language model probability for a partial hypothesis e_1 when it is expanded into a longer hypothesis by a target phrase e_2 that translates an uncovered source segment f. This expansion on the target side is similar to the monotone combination of (20) in that a newly translated phrase is concatenated to an early translated sequence.

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

$$\frac{[\mathcal{V};\mathcal{L}(e_1)] : MI(e_1) \quad e_2/f : MI(e_2)}{[\mathcal{V}';\mathcal{L}(e_1e_2)] : MI(e_1)MI(e_2)MI(e_1 \mapsto e_2)} \tag{30}$$

Fig. 7. Integrating the MI trigger model into a standard phrase-based decoder.

The MI trigger model of (10) can also be integrated into the standard phrase-based decoder as shown in Figure 7. Whenever a partial hypothesis e_1 is extended by a new target phrase e_2, we can quickly retrieve the pre-computed PMI value for each trigger pair where the triggered word locates in the newly translated target phrase e_2 and the trigger is outside the n-word window of the triggered word in e_1. This procedure is quite similar to (27) in Figure 5. And likewise, we also use the last $n-1$ words of hypotheses for hypotheses recombination.

8 Experiments

We conducted a series of experiments to evaluate the effectiveness of the proposed backward language model and MI trigger model on BTG-based SMT. Our experiments were carried out on three language pairs, namely Chinese-to-English, Spanish-to-English and Vietnamese-to-English. From these experiments, we would like to investigate the following issues.

(1) Is the backward language model competitive with the forward language model? And can we obtain improvements if we combine them together?
(2) Is the MI trigger model able to improve translation quality? If so, how can we select and construct trigger pairs so that we can maximize the improvement?
(3) Can we obtain a further improvement if we jointly apply both the backward language model and MI trigger model?

8.1 Experimental setup

In this section, we describe our experimental configurations on the three language pairs. We adopted the case-insensitive BLEU-4 (Papineni *et al.* 2002) and METEOR (Banerjee and Lavie 2005) as the evaluation metrics. In order to alleviate the impact of MERT (Och 2003) instability, we followed the suggestion of Clark *et al.* (2011) to run MERT three times and report average BLEU/METEOR scores over the three runs for all our experiments.

8.1.1 Chinese-to-english (ZH–EN)

The training corpora for Chinese-to-English are from LDC, namely LDC2004E12, LDC2004T08, LDC2005T10, LDC2003E14, LDC2002E18, LDC2005T06, LDC2003E07 and LDC2004T07. Statistics of the training corpora are shown in Table 2. We used all corpora to train our translation model and smaller corpora without the United Nations corpus (LDC2004E12) to build a maximum entropy-based reordering model (Xiong *et al.* 2006).

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

Table 2. *Parallel data sets used for Chinese/Spanish/Vietnamese-to-English experiments*

Task	No. of sentences	No. of words	
		SRC	TGT
Chinese-to-English	3.80M	96.90M	109.5M
Spanish-to-English	1.29M	27.97M	26.91M
Vietnamese-to-English	0.48M	13.07M	12.78M

To train the forward/backward language models and MI trigger model, we used the Xinhua section of the English Gigaword corpus (306 million words). Firstly, we built a forward 5-gram language model using the SRILM toolkit (Stolcke 2002) with modified Kneser– Ney smoothing. Then we trained a backward 5-gram language model on the same monolingual corpus in the way described in Section 4.2. Finally, we trained our MI trigger model still on this corpus according to the method in Section 5.2.

We used the NIST MT03 evaluation test data as the development set, and the NIST MT-04 as the test set. To optimize some configurations on the MI trigger model (Sections 8.3.1 and 8.3.2), we also used the NIST MT-05 as the test set. Each sentence in the dev/test set has four different reference translations.

8.1.2 Spanish-to-english (ES–EN)

For experiments on Spanish-to-English, we used data from the *ACL 2010 Joint Workshop on Statistical Machine Translation and MetricsMATR* (WMT10) (Callison-Burch *et al.* 2010). In particular, we employed the News Commentary and Europarl v5 (Koehn 2005) as our parallel training data. Following the maximum sentence length setting that was adopted by the baseline system and many other participant systems in the workshop, we removed sentence pairs of more than 40 words either in the source side or the target side. Details are given in Table 2.

To train our 5-gram forward/backward language models and MI trigger model, we used the News Commentary10 and Europarl v5 English corpora. These corpora contain 45.28M words in 1.88M sentences. Note that we did not utilize the English News-shuffled corpus in the language model and MI trigger model training because the incorporation of such a huge corpus in training is beyond our current computational ability.

The 2008 test set was used as the development set and the 2010 test set as our test set. All training corpora and dev/test sets were lowercased and tokenized with tools from the workshop. Each sentence in the dev/test set has only one reference translation.[6]

[6] This might be one of reasons that the improvements on ES–EN obtained by the two language models are normally lower than those on ZH–EN and VI–EN.

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

Table 3. *Effects of the proposed backward language model. FLM: forward language model; BLM: backward language model; MR: METEOR*

Model	ZH–EN		ES–EN		VI–EN	
	BLEU	MR	BLEU	MR	BLEU	MR
FLM (baseline)	35.62	53.65	25.54	53.46	30.06	56.27
BLM	36.12	54.12	25.75	53.61	30.63	56.52
FLM + BLM	36.22	54.21	25.81	53.60	30.54	56.54

8.1.3 Vietnamese-to-english (VI–EN)

The training corpora for the Vietnamese-to-English translation task are from our internal project. We collected about 500K sentence pairs (as shown in Table 2) in news domain. In this task, we adopted the language models and the MI trigger model trained in the Chinese-to-English task. We also collected 1,117 sentences as the development set and 1,000 sentences as the test set. Each sentence in the dev/test set has four different reference translations.

8.2 Backward language model

We investigate the effectiveness of the proposed backward language model within two contexts: (1) using it alone and (2) combining it with the forward language model.

8.2.1 Backward language model alone

As we discussed in Section 6.1, we can completely replace the forward language model with the backward language model. The only difference is that the order of input to the backward language model is reversed. Table 3 shows the results of using the backward language model alone across the three language pairs. Compared with the forward language model, the backward language model achieves better performance in terms of both BLEU and METEOR. This resonates with the finding of Finch and Sumita (2009), who report that among translations from sixteen different source languages to English, reverse decoding (with the backward language model) outperforms forward decoding (with the forward language model) in most cases (13 versus 3). The results in Table 3 empirically reconfirm that the backward language model may be a better choice than the forward language model when a source language is translated to English.

The reason why the backward language model generally outperforms the forward language model in X-to-English translation may be related to the distribution of head word positions relative to current words in English. We study such distribution on the English Penn Treebank (Marcus, Santorini and Marcinkiewicz 1993) where all parse trees are automatically transferred to dependency trees[7]. We assume that the

[7] Penn2Malt is used to produce dependency trees from the English Treebank.

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

Table 4. *Perplexity of the forward/backward language models on the reference sets*

Model	ZH–EN	ES–EN	VI–EN
FLM	159.8	377.8	482.6
BLM	154.3	381.1	478.4

head of a dependency relation is more predictive of its dependent objects than vice versa. In the dependency structure assigned to a sentence, all words are dependent on some head, except for the matrix verb. Analysis of the dependency structures in the corpus shows that 86.43% of words have their head within a 5-word window to their left or to their right. Of those, 53.3% have their head within a 5-word window to their RIGHT. Hence the majority of words depend on a head that *follows* them, rather than on a head that *precedes* them. In other words, most words have backward (rather than forward) dependencies, which endows the backward language model with a stronger prediction ability than that of the forward language model in English.

In view of the difficulty in the syntactical dependency analysis, here we provide a more practical and straightforward indicator to show whether we should use a backward language model. As we can train a backward or forward language model on the same training corpus, we can also easily calculate the perplexity of them on the same test corpus. Table 4 displays the perplexity of the forward/backward language models on the reference corpora of the test sets of three language pairs. The backward language model has a lower perplexity on ZH/VI–EN. Given the higher performance of the backward language model on ZH/VI–EN shown in Table 3, the lower perplexity of the BLM is a strong indicator that we should incorporate the backward language model.

8.2.2 Backward and forward language model combination

The last row of Table 3 shows that the integration of a forward language model and a backward language model achieves improvements in terms of both BLEU and METEOR over a single forward language model across all three language pairs. The absolute improvements in BLEU score are 0.60, 0.27, and 0.48 BLEU points on ZH/ES/VI–EN, respectively. The relatively lower improvements on Spanish-to-English may be related to the higher perplexity of the backward language model on the reference set of ES–EN as shown in Table 4.

Table 5 provides a Chinese-to-English example that displays how the backward language model improves translation quality. The italicized words in the hypothesis generated by using the backward and forward language model together (FLM + BLM) exactly match the reference. However, the italicized words in the baseline hypothesis fail to match the reference due to the incorrect position of the word ‘decree’ (XXXX). We calculate the forward/backward language model score (the logarithm of language model probability) for the italicized words in both the baseline and FLM + BLM hypothesis according to the trained language models. The

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

Table 5. *Translation example from the NIST MT-04 test set, comparing the baseline with the backward language model*

Source	北京 青年报 报导 , 北京 农业局 最近 发出 一连串 的 防治 及 监督 法令
Baseline	Beijing Youth Daily reported that Beijing agricultural decree recently issued a series of control and supervision
FLM + BLM	Beijing Youth Daily reported that Beijing Bureau of Agriculture recently issued a series of prevention and control laws
Reference	Beijing Youth Daily reported that Beijing Bureau of Agriculture recently issued a series of preventative and monitoring ordinances

difference in the forward language model score is only 1.582, which may be offset by differences in other features in the log-linear translation model. On the other hand, the difference in the backward language model score is 3.517. This larger difference may guarantee that the hypothesis generated by FLM + BLM is better enough to be selected as the best hypothesis by the decoder. This suggests that the backward language model is able to provide useful and discriminative information which is complementary to that given by the forward language model.

8.3 MI trigger model

In this section, we integrate various trigger pairs into our MI trigger model and thoroughly investigate their impact on translation quality in terms of BLEU/METEOR. In particular,

(1) We study the distribution of trigger pairs according to their PMI values and component words. We then explore different trigger pair selection criteria based on this distribution.
(2) We further investigate whether we should use distance-dependent or syntactically informed triggers.
(3) We finally evaluate our MI trigger model on the three language pairs.

Note that all experiments of (1) and (2) are conducted on the Chinese-to-English translation task with the NIST MT-05 being the test set.

8.3.1 Trigger pair selection

Trigger pairs (x, y) can be divided into three categories according to their PMI value $PMI(x, y)$.

(1) $PMI(x, y) > 0$, if there is a genuine association between x and y (Church and Hanks 1990).
(2) $PMI(x, y) = 0$, if x and y occur independently.

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

Table 6. *Distribution of trigger pairs on the Chinese-to-English translation task*

Incl. stop words	PMI Pos.	PMI Neg.	Total
Yes	18.93%	24.64%	43.57%
No	38.43%	18%	56.43%
Total	57.36%	42.64%	100%

Table 7. *Results of using various trigger pair selection criteria*

Model	BLEU	METEOR
FLM (baseline)	34.28	55.04
All	34.02	55.18
Pos	34.73	55.30
Pos-sw	34.65	54.97
Neg	33.81	54.68

(3) $PMI(x, y) < 0$, if x and y are in complementary distribution[8] (Church and Hanks 1990) or (x, y) include words which occur with a very high frequency, such as stop words.

We investigate the distribution of such categories on the Chinese-to-English translation task. We collect 5.03M trigger pairs in total from the training data after we remove trigger pairs that violate the selection criteria (1) and (2) described in Section 5.2. Table 6 gives the distribution on these collected trigger pairs.

From the table, we can observe that the majority of trigger pairs (accounting for 57.36%) are with a positive PMI value. Among these PMI-positive trigger pairs, most of them (70%) do not contain any stop words. However, among PMI-negative trigger pairs, most of them (accounting for 57.79%) include stop words. Based on this distribution, we conducted several experiments on the NIST MT-05: (1) using all trigger pairs (All); (2) only using PMI-positive trigger pairs (Pos); (3) only using PMI-positive trigger pairs without stop words (Pos-sw); and finally (4) only using PMI-negative trigger pairs (Neg). The results are shown in Table 7.

We can see from the table that we achieve the best performance in terms of BLEU when we only integrate PMI-positive trigger pairs into the MI trigger model (Pos). When we exploit both PMI-positive and PMI-negative trigger pairs (All), the BLEU score is significantly dropped (34.73 versus 34.02) compared with the BLEU score of Pos. Furthermore, only using PMI-negative trigger pairs (Neg) leads to a more seriously degraded performance which is worse than that of the baseline. This indicates that PMI-negative trigger pairs may not help even if they provide complementary cues.

Yet another interesting finding is about trigger pairs with stop words. If we remove PMI-positive trigger pairs that contain stop words (Pos-sw), the result shows that

[8] If x occurs, y should not occur in the same environment.

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

Table 8. *Results of using distance-dependent/syntactically informed MI triggers. MI-di: distance-independent MI model; MI-dd: distance-dependent MI model; MI-syn: syntactically informed MI model*

Model	BLEU	METEOR
FLM (baseline)	34.28	55.05
FLM + MI-di	34.73	55.30
FLM + MI-dd	34.56	54.37
FLM + MI-syn	34.36	55.09

the absence of trigger pairs with stop words degrades the performance in terms of BLEU and METEOR. This suggests that stop words are important in constructing trigger pairs. Section 8.3.3 gives an example of a trigger pair that contains a stop word 'is' and shows how this trigger pair helps the decoder find a better hypothesis.

In all experiments hereafter, we therefore only use PMI-positive trigger pairs (including PMI-positive trigger pairs with stop words) in our MI trigger model.

8.3.2 Distance-dependent triggers and syntactically informed triggers

In this section, we explore two informative triggers: distance-dependent triggers and syntactically informed triggers. Our goal is to find whether the associated distance and syntax information are able to improve our MI trigger model. In the distance-dependent MI trigger model, we maintain the distance between two words w_k and w_i as the third element in each trigger, following Zhou's model (Zhou 2004). The PMI of a distance-dependent trigger is therefore calculated as follows

$$PMI(w_k, w_i, i-k-1) = \log\left(\frac{P(w_k, w_i, i-k-1)}{P(w_k)P(w_i)}\right) \tag{26}$$

Table 8 shows the result of the distance-dependent MI trigger model on the NIST MT-05. Compared with the distance-independent MI trigger model, the distance-dependent model obtains a degraded performance in terms of both BLEU and METEOR, especially the latter which is even worse than that of the baseline. Therefore, we do not adopt the distance-dependent model given its larger model size with a worse performance compared with the distance-independent model.

In order to employ syntactical information, we annotated each word in the training data with part-of-speech (POS) tags via a conditional random field (CRF) based POS tagger.[9] The extracted trigger pairs $(w_k/t_k, w_i/t_i)$ are therefore associated with POS information (t_k and t_i). Words in extracted bilingual phrases are also associated with POS tags so that we can obtain POS tags for target words during decoding. With the remaining (training procedure and trigger selection) kept the same as the distance-independent model, the syntactically informed MI trigger model is not able to achieve further improvements either in terms of BLEU and METEOR as shown in Table 8.

[9] http://sourceforge.net/projects/crftagger/

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

Table 9. *Results of combining the MI trigger model with the forward and backward language model. For the convenience of comparison, here we repeat the results of FLM + BLM from Table 3. MR: METEOR*

	ZH–EN		ES–EN		VI–EN	
Model	BLEU	MR	BLEU	MR	BLEU	MR
FLM (baseline)	35.62	53.65	25.54	53.46	30.06	56.27
FLM + BLM	36.22	54.21	25.81	53.60	30.54	56.54
FLM + MI	36.01	54.02	26.00	53.65	30.50	56.58
FLM + BLM + MI	36.59	53.83	25.94	53.65	30.70	56.72

8.3.3 Effects of MI trigger model across language pairs

We evaluated our distance-independent MI trigger model on the three language pairs. Only PMI-positive trigger pairs are used as described in Section 8.3.1. The results are given in Table 9. From the table, we can see that the MI trigger model is able to improve translation quality on all three language pairs in terms of both BLEU and METEOR. We obtain absolute improvements of 0.39, 0.46, and 0.44 BLEU points on ZH–EN, ES–EN and VI–EN, respectively. This indicates that language models in SMT can significantly benefit from triggers which carry long-distance dependency information.

In Table 10, we present a Chinese-to-English example to show how the MI trigger model improves translation quality. The major difference in hypotheses of this example is the word choice between 'is' and 'was'. The new system enhanced with the MI trigger model (FLM + MI) selects the former while the baseline selects the latter. The forward language model score for the baseline hypothesis is −26.4141, which is higher than the forward language model score of the FLM + MI hypothesis −26.6698. This could be the reason why the baseline selects the word 'was' instead of 'is'. As can be seen, there is another 'is' in the preceding context of the word 'was' in the baseline hypothesis. Unfortunately, this word 'is' is located just outside the scope of the preceding 5-gram context of 'was'. The forward 5-gram language model is hence not able to take it into account when calculating the probability of 'was'. However, this is not a problem for the MI trigger model. Since 'is' and 'was' rarely co-occur in the same sentence, the PMI value of the trigger pair (is, was)[10] is −1.03114 while the PMI value of the trigger pair (is, is) is as high as 0.315516. Therefore, our MI trigger model selects 'is' rather than 'was'.[11] This example illustrates that the MI trigger model is capable of selecting correct words by using long-distance trigger pairs.

[10] Since we remove all trigger pairs whose PMI value is negative, the PMI value of this pair (is, was) is set 0 in practice in the decoder.

[11] The overall MI trigger model scores (the logarithm of (10)) of the baseline hypothesis and the FLM + MI hypothesis are 2.087 and 2.251, respectively.

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

Table 10. *Translation example from the NIST MT-04 test set, comparing the baseline with the MI trigger model. Both system outputs are not detokenized so that we can see how language model scores are calculated. The underlined words highlight the difference between the enhanced models and the baseline*

Source	自卫队 此行 之所以 引人瞩目 , 是 因为 它 并非 是 一个 孤立 的 事件 。
Baseline	Self-Defense Forces' trip is remarkable, because it <u>was</u> not an isolated incident.
FLM + MI	Self-Defense Forces' trip is remarkable, because it <u>is</u> not an isolated incident.
Reference	Self-Defense Forces' trip arouses attention, because it <u>is</u> not an isolated incident.

Table 11. *Relative weights of the backward language model (BLM) and the mutual information trigger model (MI) against the weight of the forward language model assigned by MERT. We set the weights of the forward language model as the reference (values are set to 1)*

	FLM	BLM	MI
ZH–EN	1	3.23	0.26
ES–EN	1	0.13	0.33
VI–EN	1	1.36	0.34

8.4 Combination of the backward language model and MI trigger model

The last question is whether a combination of the backward language model and the MI trigger model is able to gain further improvements. According to the results shown in Table 9, the answer to this question is mixed. In Chinese/Vietnamese-to-English, the combination of the backward language model and the MI trigger model (FLM + BLM + MI) is able to obtain improvements of 0.97 and 0.64 BLEU points, respectively, over the single forward language model. These improvements are larger than that achieved by integrating only one model with the forward language model (FLM + BLM or FLM + MI). In Spanish-to-English, we successfully achieve further improvements over FLM + BLM but fail to obtain further improvements over FLM + MI. It seems that the MI trigger model is more helpful than the backward language model in Spanish-to-English translation.

This is also suggested by the relative weights assigned by MERT to the backward and MI trigger language models against the weight of the forward language model, which are shown in Table 11. In Chinese/Vietnamese-to-English, the backward language model has a much larger weight than that of the forward language model. This indicates that the tuning algorithm prefers the backward language model to the forward language model in these two tasks. However, in the Spanish-to-English translation task, the backward language model obtains a very low weight. Such relative weights of the forward/backward language models are exactly consistent with the perplexities of these two language models shown in Table 4. Similarly, the

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

Table 12. *Language model size and decoding speed*

Model	Language model size	Decoding speed
Baseline (FLM)	1	1
BLM	0.91	0.92
FLM + BLM	1.91	1.06
FLM + MI	1.12	3.84
FLM + BLM + MI	2.03	3.64

relative weights of the MI trigger model are also in line with its contributions in the translation tasks of the three language pairs.

8.5 Language model size and decoding speed

In this section, we discuss the size of three language models as well as decoding speed with these models. Table 12 shows the information of model size and decoding speed in the Chinese–English translation task. In order to have a clear comparison, we set the model size and decoding speed of the baseline system (with the forward language model) as the reference (values are set to 1). Then we show the ratios of model size and decoding speed of other systems against those of the baseline. From the table, we can observe that

- The size of the backward language model is marginally smaller than that of the forward language model (maybe due to smoothing).
- The size of the MI trigger model is quite small, about 10% of that of the forward language model.
- The decoder with the backward language model is faster than that with the forward language model (maybe due to the smaller model size). The combination of two language models (BLM + FLM) does not slow the decoding quite much (only 6% slower than the baseline). This displays the advantage of our integration algorithm as we do not need to run the decoding process twice, each of which is with a different language model (forward or backward).
- The decoding speed for systems with the MI trigger model is much slower than the baseline. The reason for this is that we do not use an efficient data structure to access the huge trigger pair table. We store trigger pairs in an ordinary standard map. We believe that the decoding speed will be much faster if we do some optimizations with a more efficient data structure, such as trie.

9 Conclusion and further work

We have presented two models to enhance the ability of standard *n*-gram language models in capturing richer contexts and long-distance dependencies that go beyond the scope of forward *n*-gram windows. The first model is the backward language model which uses backward *n*-grams to predict the current word. The second model is the MI trigger model that incorporates long-distance trigger pairs into language

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

modeling. We introduced algorithms that directly integrate the two models into a BTG-based CKY-style decoder and a standard phrase-based decoder, respectively.

We conducted large-scale experiments on Chinese/Spanish/Vietnamese-to-English to thoroughly evaluate the effectiveness of the two proposed models. Our experimental results validate that

(1) The backward language model is competitive with the forward language model and significantly outperforms the forward language model in Chinese/Vietnamese-to-English translation.
(2) The joint of the backward and forward language models significantly outperforms the single forward language model on all three language pairs.
(3) The MI trigger model which employs only PMI-positive trigger pairs is able to improve the performance significantly.
(4) The combination of the backward language model and the MI trigger model achieves further improvements in Chinese/Vietnamese-to-English translation.

Although BTG is fifteen-year-old synchronous grammar technology, it is still very competitive for SMT. Recent experiments reveal that the performance of BTG-based phrasal SMT with lexicon-sensitive reordering (Xiong *et al.* 2006) is comparable with that of the hierarchical phrase-based system (Chen *et al.* 2008; He *et al.* 2008). This is the reason that we currently use the BTG-based system to validate the effectiveness of our method. In the future, we plan to integrate the backward and trigger language models into hierarchical phrase-based and syntax-based SMT systems. We believe that these systems will also benefit from backward *n*-grams and triggers.

References

Banerjee, S., and Lavie, A. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann Arbor, MI, pp. 65–72.

Brants, T., Popat, A. C., Xu, P., Och, F. J., and Dean, J. 2007. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, pp. 858–67.

Callison-Burch, C., Koehn, P., Monz, C., Peterson, K., Przybocki, M., and Zaidan, O. 2010. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, Uppsala, Sweden, pp. 17–53.

Charniak, E., Knight, K., and Yamada, K. 2003. Syntax-based language models for statistical machine translation. In *Proceedings of MT Summit IX*, New Orleans, USA, pp. 40–46.

Chen, B., Xiong, D., Zhang, M., Aw, A., and Li, H. 2008. I2r multi-pass machine translation system for iwslt 2008. In *Proceeding of the International Workshop on Spoken Language Translation 2008*, Hawaii, USA, pp. 46–51.

Chiang, D. 2007. Hierarchical phrase-based translation. *Computational Linguistics* **33**(2): 201–28.

Church, K. W., and Hanks, P. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics* **16**(1):22–9.

Clark, J. H., Dyer, C., Lavie, A., and Smith, N. A.. 2011. Better hypothesis testing for statistical machine translation: controlling for optimizer instability. In *Proceedings of the 49th Annual*

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, Oregon, USA, pp. 176–81.

Duchateau, J., Demuynck, K., and Wambacq, P. 2002. Confidence scoring based on backward language models. In *Proceedings of ICASSP*, Orlando, FL, pp. 221–4.

Emami, A., Papineni, K., and Sorensen, J. 2007. Large-scale distributed language modeling. In *Proceedings of ICASSP*, Honolulu, HI, pp. 37–40.

Finch, A., and Sumita, E. 2009. Bidirectional phrase-based statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, pp. 1124–32.

He, X., Yang, M., Gao, J., Nguyen, P., and Moore, R. 2008. Indirect-HMM-based hypothesis alignment for combining outputs from machine translation systems. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Honolulu, Hawaii, pp. 98–107.

Koehn, P. 2005. Europarl: a parallel corpus for statistical machine translation. In *the tenth Machine Translation Summit*, Phuket, Thailand, pp. 79–86.

Koehn, P., Och, F. J., and Marcu, D. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Edmonton, Canada, pp. 58–54.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. 1993. Building a large annotated corpus of English: the penn treebank. *Computational Linguistics* **19**(2): 313–30.

Mauser, A., Hasan, S., and Ney, H. 2009. Extending statistical machine translation with discriminative and trigger-based lexicon models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, pp. 210–18.

Och, F. J. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, pp. 160–7.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA, USA, pp. 311-18.

Post, M., and Gildea, D. 2008. Parsers as language models for statistical machine translation. In *Proceedings of AMTA*, Waikiki, Hawai'i, pp. 172–181.

Raybaud, S., Lavecchia, C., Langlois, D., and Smaïli, K. 2009. New confidence measures for statistical machine translation. In *Proceedings of the International Conference on Agents and Artificial Intelligence*, Porto, Portugal, pp. 61–8.

Rosenfeld, R., Carbonell, J., and Rudnicky, A. 1994. Adaptive statistical language modeling: a maximum entropy approach. Technical Report, Carnegie Mellon University.

Shen, L., Xu, J., and Weischedel, R. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-08: HLT*, Columbus, Ohio, pp. 577–85.

Stolcke, A. 2002. Srilm–an extensible language modeling toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing*, Denver, Colorado, USA, pp. 901–4.

Talbot, D., and Osborne, M. 2007. Randomised language modelling for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, pp. 512-19.

Wu, D. 1996. A polynomial-time algorithm for statistical machine translation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, Santa Cruz, California, USA, pp. 152–8.

Wu, D. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics* **23**(3): 377–403.

Xiong, D., Liu, Q., and Lin, S. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on*

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press

Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia, pp. 521–8.

Xiong, D., Zhang, M., and Li, H. 2011. Enhancing language models in statistical machine translation with backward n-grams and mutual information triggers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA, pp. 1288–97.

Zhang, Y., Hildebrand, A. S., and Vogel, S. 2006. Distributed language modeling for *n*-best list re-ranking. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, pp. 216–23.

Zhou, G. 2004. Modeling of long distance context dependency. In *Proceedings of Coling*, Geneva, Switzerland, pp. 92–8.

https://doi.org/10.1017/S1351324913000168 Published online by Cambridge University Press