Enhancing Language Models in Statistical Machine Translation with Backward N-grams and Mutual Information Triggers

Deyi Xiong, Min Zhang, Haizhou Li
Human Language Technology
Institute for Infocomm Research
1 Fusionopolis Way, #21-01 Connexis, Singapore 138632
{dyxiong, mzhang, hli}@i2r.a-star.edu.sg

Abstract

In this paper, with a belief that a language model that embraces a larger context provides better prediction ability, 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 integrate the two proposed models into phrase-based statistical machine translation and conduct experiments on large-scale training data to investigate their effectiveness. Our experimental results show that both models are able to significantly improve translation quality and collectively achieve up to 1 BLEU point over a competitive baseline.

1 Introduction

Language model is one of the most important knowledge sources for statistical machine translation (SMT) (Brown et al., 1993). The standard n-gram language model (Goodman, 2001) assigns probabilities to hypotheses in the target language conditioning on a context history of the preceding $n-1$ words. Along with the efforts that advance translation models from word-based paradigm to syntax-based philosophy, in recent years we have also witnessed increasing efforts dedicated to extend standard n-gram language models for SMT. We roughly categorize these efforts into two directions: data-volume-oriented and data-depth-oriented.

In the first direction, more data is better. In order to benefit from monolingual corpora (LDC news data or news data collected from web pages) that consist of billions or even trillions of English words, huge language models are built in a distributed manner (Zhang et al., 2006; Brants et al., 2007). Such language models yield better translation results but at the cost of huge storage and high computation.

The second direction digs deeply into monolingual data to build linguistically-informed language models. For example, Charniak et al. (2003) present a syntax-based language model for machine translation which is trained on syntactic parse trees. Again, Shen et al. (2008) explore a dependency language model to improve translation quality. To some extent, these syntactically-informed language models are consistent with syntax-based translation models in capturing long-distance dependencies.

In this paper, we pursue the second direction without resorting to any linguistic resources such as a syntactic parser. With a belief that a language model that embraces a larger context provides better prediction ability, we learn additional information from training data to enhance conventional n-gram language models and extend their ability to capture richer contexts and long-distance dependencies. In particular, we integrate backward n-grams and mutual information (MI) triggers into language models in SMT.

In conventional n-gram language models, we 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

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 1288–1297,
Portland, Oregon, June 19-24, 2011.

on forward n-grams as 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 the decoder. In doing so, we attempt 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 previous contexts, which however go beyond the scope of forward n-grams. If the current word is indexed as w_i, the farthest word that the forward n-gram 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}. By 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 proposed backward language model and the MI trigger model into a state-of-the-art phrase-based SMT system. We evaluate the effectiveness of both models on Chinese-to-English translation tasks with large-scale training data. Compared with the baseline which only uses the forward language model, our experimental results show that the additional backward language model is able to gain about 0.5 BLEU points, while the MI trigger model gains about 0.4 BLEU points. When both models are integrated into the decoder, they collectively improve the performance by up to 1 BLEU point.

The paper is structured as follows. In Section 2, we will briefly introduce related work and show how our models differ from previous work. Section 3 and 4 will elaborate the backward language model and the MI trigger model respectively in more detail, describe the training procedures and explain how the models are integrated into the phrase-based decoder. Section 5 will empirically evaluate the effectiveness of these two models. Section 6 will conduct an in-depth analysis. In the end, we conclude in Section 7.

2 Related Work

Previous work devoted to improving language models in SMT mostly focus on two categories as we mentioned before[1]: large language models (Zhang et al., 2006; Emami et al., 2007; Brants et al., 2007; Talbot and Osborne, 2007) and syntax-based language models (Charniak et al., 2003; Shen et al., 2008; Post and Gildea, 2008). Since our philosophy is fundamentally different from them in that we build contextually-informed language models by using backward n-grams and MI triggers, we discuss previous work that explore these two techniques (backward n-grams and MI triggers) in this section.

Since the context "history" in the backward language model (BLM) is actually the future words to be generated, BLM is normally used in a post-processing where all words have already been generated or in a scenario where sentences are proceeded from the ending to the beginning. Duchateau et al. (2002) use the BLM score as a confidence measure to detect wrongly recognized words in speech recognition. Finch and Sumita (2009) use the BLM in their reverse translation decoder where source sentences are proceeded from the ending to the beginning. Our BLM is different from theirs in that we access the BLM during decoding (rather than after decoding) where source sentences are still proceeded from the beginning to the ending.

Rosenfeld et al. (1994) introduce trigger pairs into a maximum entropy based language model as features. The trigger pairs are selected according to their mutual information. Zhou (2004) also propose 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 mutual information of distance-dependent trigger pairs. Our MI trigger model is mostly inspired by the work of these two papers, especially by Zhou's MI-Ngram model (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 in the MI calculation and trigger pair selection. Mauser et al. (2009) propose bilingual triggers where two source words trigger one target word to

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

improve lexical choice of target words. Our analysis (Section 6) show that our monolingual triggers can also help in the selection of target words.

3 Backward Language Model

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

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

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 $w_{i-n+1}...w_{i-1}$ instead of the whole context history $w_1...w_{i-1}$.

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

$$P_b(w_1^m) = \prod_{i=1}^{m} P(w_i|w_{i+1}^{m}) \approx \prod_{i=1}^{m} P(w_i|w_{i+1}^{i+n-1}) \quad (2)$$

3.1 Training

For the convenience of training, we invert the order in each sentence in the training data, i.e., from the original order $(w_1...w_m)$ to the reverse order $(w_m...w_1)$. 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 Eq. (2) is $w_{i+n-1}...w_{i+1}$ instead of $w_{i+1}...w_{i+n-1}$ after we invert the order. The words are the same but the order is completely reversed.

3.2 Decoding

In this section, we will present two algorithms to integrate the backward n-gram language model into two kinds of phrase-based decoders respectively: 1) a CKY-style decoder that adopts bracketing transduction grammar (BTG) (Wu, 1997; Xiong et al., 2006) and 2) a standard phrase-based decoder (Koehn et al., 2003). Both decoders translate source sentences from the beginning of a sentence to the ending. Wu (1996) introduce 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 CFGs by Chiang (2007). Our algorithms are different from theirs in two major aspects

1. The string input to the algorithms is in a reverse order.
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 algorithms, we define three functions $\mathcal{P}$, $\mathcal{L}$, and $\mathcal{R}$ on strings (in a reverse order) over the English terminal alphabet T. The function $\mathcal{P}$ is defined as follows.

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

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} ($P(w_{k-n+2}|w_k...w_{k-n+3})$). 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. Since

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

we calculate backward language model probabilities during a beginning-to-ending (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 $\mathcal{P}$ 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 $\mathcal{L}$ and $\mathcal{R}$ are defined as follows

$$\mathcal{L}(w_k...w_1) = \begin{cases} w_k...w_{k-n+2}, & \text{if } k \geq n \\ w_k...w_1, & \text{otherwise} \end{cases} \quad (4)$$

$$\mathcal{R}(w_k...w_1) = \begin{cases} w_{n-1}...w_1, & \text{if } k \geq n \\ w_k...w_1, & \text{otherwise} \end{cases} \quad (5)$$

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

Following Chiang (2007), we describe our algorithms in a deductive system. We firstly show the algorithm[3] that integrates the backward language model into a BTG-style decoder (Xiong et al., 2006) in Figure 1. The item $[A, i, j; l|r]$ indicates that a BTG node A 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. As mentioned before, all target strings assessed by the defined functions ($\mathcal{P}$, $\mathcal{L}$, and $\mathcal{R}$) are in an inverted order (denoted by $\overline{e}$). We only display the backward language model probability for each item, ignoring all other scores such as phrase translation probabilities. The Eq. (8) in Figure 1 shows how we calculate the backward language model probability for the axiom which applies a BTG lexicon rule to translate a source phrase c into a target phrase e. The Eq. (9) and (10) show how we update the backward language model probabilities for two inference rules which combine two neighboring blocks in a straight and inverted order respectively. The fundamental theories behind this update are

$$\mathcal{P}(\overline{e_1 e_2}) = \mathcal{P}(\overline{e_1})\mathcal{P}(\overline{e_2})\frac{\mathcal{P}(\mathcal{R}(\overline{e_2})\mathcal{L}(\overline{e_1}))}{\mathcal{P}(\mathcal{R}(\overline{e_2}))\mathcal{P}(\mathcal{L}(\overline{e_1}))} \quad (6)$$

[3]It can also be easily adapted to integrate the forward n-gram language model.

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

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

$$\mathcal{P}(\overline{e_2 e_1}) = \mathcal{P}(\overline{e_1})\mathcal{P}(\overline{e_2})\frac{\mathcal{P}(\mathcal{R}(\overline{e_1})\mathcal{L}(\overline{e_2}))}{\mathcal{P}(\mathcal{R}(\overline{e_1}))\mathcal{P}(\mathcal{L}(\overline{e_2}))} \quad (7)$$

Whenever two strings e_1 and e_2 are concatenated in a straight or inverted order, we can reuse their $\mathcal{P}$ values ($\mathcal{P}(\overline{e_1})$ and $\mathcal{P}(\overline{e_2})$) in terms of dynamic programming. Only the probabilities of boundary words (e.g., $\mathcal{R}(\overline{e_2})\mathcal{L}(\overline{e_1})$ in Eq. (6)) need to be recalculated since they have complete n-grams after the concatenation. Table 1 shows values of $\mathcal{P}$, $\mathcal{L}$, and $\mathcal{R}$ in a 3-gram example which helps to verify Eq. (6). These two equations guarantee that our algorithm can correctly compute the backward language model probability of a sentence stepwise in a dynamic programming framework.[4]

The theoretical time complexity of this algorithm is $\mathcal{O}(m^3|T|^{4(n-1)})$ because in the update parts in Eq. (6) and (7) both the numerator and denominator have up to $2(n-1)$ terminal symbols. This is the same as the time complexity of Chiang's language model integration (Chiang, 2007).

Figure 2 shows the algorithm that integrates the backward language model into a standard phrase-based SMT (Koehn et al., 2003). $\mathcal{V}$ denotes a coverage vector which records source words translated so far. The Eq. (11) shows how we update the backward language model probability for a partial hypothesis when it is extended into a longer hypothesis by a target phrase translating an uncovered source

[4]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.

$$\frac{A \rightarrow c/e}{[A, i, j; \mathcal{L}(\overline{e})|\mathcal{R}(\overline{e})] : \mathcal{P}(\overline{e})} \tag{8}$$

$$\frac{A \rightarrow [A_1, A_2] \quad [A_1, i, k; \mathcal{L}(\overline{e_1})|\mathcal{R}(\overline{e_1})] : \mathcal{P}(\overline{e_1}) \quad [A_2, k+1, j; \mathcal{L}(\overline{e_2})|\mathcal{R}(\overline{e_2})] : \mathcal{P}(\overline{e_2})}{[A, i, j; \mathcal{L}(\overline{e_1 e_2})|\mathcal{R}(\overline{e_1 e_2})] : \mathcal{P}(\overline{e_1})\mathcal{P}(\overline{e_2})\frac{\mathcal{P}(\mathcal{R}(\overline{e_2})\mathcal{L}(\overline{e_1}))}{\mathcal{P}(\mathcal{R}(\overline{e_2}))\mathcal{P}(\mathcal{L}(\overline{e_1}))}} \tag{9}$$

$$\frac{A \rightarrow \langle A_1, A_2 \rangle \quad [A_1, i, k; \mathcal{L}(\overline{e_1})|\mathcal{R}(\overline{e_1})] : \mathcal{P}(\overline{e_1}) \quad [A_2, k+1, j; \mathcal{L}(\overline{e_2})|\mathcal{R}(\overline{e_2})] : \mathcal{P}(\overline{e_2})}{[A, i, j; \mathcal{L}(\overline{e_2 e_1})|\mathcal{R}(\overline{e_2 e_1})] : \mathcal{P}(\overline{e_1})\mathcal{P}(\overline{e_2})\frac{\mathcal{P}(\mathcal{R}(\overline{e_1})\mathcal{L}(\overline{e_2}))}{\mathcal{P}(\mathcal{R}(\overline{e_1}))\mathcal{P}(\mathcal{L}(\overline{e_2}))}} \tag{10}$$

Figure 1: Integrating the backward language model into a BTG-style decoder.

$$\frac{[\mathcal{V}; \mathcal{L}(\overline{e_1})] : \mathcal{P}(\overline{e_1}) \quad c/e_2 : \mathcal{P}(\overline{e_2})}{[\mathcal{V}'; \mathcal{L}(\overline{e_1 e_2})] : \mathcal{P}(\overline{e_1})\mathcal{P}(\overline{e_2})\frac{\mathcal{P}(\mathcal{R}(\overline{e_2})\mathcal{L}(\overline{e_1}))}{\mathcal{P}(\mathcal{R}(\overline{e_2}))\mathcal{P}(\mathcal{L}(\overline{e_1}))}} \tag{11}$$

Figure 2: Integrating the backward language model into a standard phrase-based decoder.

segment. This extension on the target side is similar to the monotone combination of Eq. (9) in that a newly translated phrase is concatenated to an early translated sequence.

4 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 work such as (Rosenfeld et al., 1994) and (Zhou, 2004) exploit trigger pairs. A trigger pair is defined as an ordered 2-tuple (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[5].

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

$$PMI(x, y) = \log(\frac{P(x, y)}{P(x)P(y)}) \tag{12}$$

[5]In this paper, we require that word x and y occur in the same sentence.

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

$$P(w_1^m) \approx (\prod_{i=1}^{m} P(w_i|w_{i-n+1}^{i-1})) \times \prod_{i=n+1}^{m} \prod_{k=1}^{i-n} \exp(PMI(w_k, w_i, i-k-1)) \tag{13}$$

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 multiples 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 paper, for simplicity, we adopt a distance-independent MI trigger model as follows

$$MI(w_1^m) = \prod_{i=n+1}^{m} \prod_{k=1}^{i-n} \exp(PMI(w_k, w_i)) \tag{14}$$

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. By 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.

4.1 Training

We can use the maximum likelihood estimation method to calculate PMI for each trigger pair by taking counts from training data. Let $C(x, y)$ be the co-occurrence count of the trigger pair (x, y) in the training data. The joint probability of (x, y) is calculated as

$$P(x, y) = \frac{C(x, y)}{\sum_{x,y} C(x, y)} \tag{15}$$

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

$$P(x) = \sum\nolimits_{y} P(x, y) \tag{16}$$

$$P(y) = \sum\nolimits_{x} P(x, y) \tag{17}$$

Since the number of distinct trigger pairs is $\mathcal{O}(|T|^2)$, 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 $x \in \{w_1 ... w_{i-5}\}$.

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.

4.2 Decoding

The MI trigger model of Eq. (14) can be directly integrated into the decoder. For the standard phrase-based decoder (Koehn et al., 2003), whenever a partial hypothesis is extended by a new target phrase, we can quickly retrieve the pre-computed PMI value for each trigger pair where the triggered word locates in the newly translated target phrase and the trigger is outside the n-word window of the triggered word. It's a little more complicated to integrate the MI trigger model into the CKY-style phrase-based decoder. But we still can handle it by dynamic programming as follows

$$MI(e_1 e_2) = MI(e_1)MI(e_2)MI(e_1 \rightarrow e_2) \tag{18}$$

where $MI(e_1 \rightarrow e_2)$ represents the PMI values in which a word in e_1 triggers a word in e_2. It is defined as follows

$$MI(e_1 \rightarrow e_2) = \prod_{w_i \in e_2} \prod_{\substack{w_k \in e_1 \\ i-k \geq n}} \exp(PMI(w_k, w_i)) \tag{19}$$

5 Experiments

In this section, we conduct large-scale experiments on NIST Chinese-to-English translation tasks to evaluate the effectiveness of the proposed backward language model and MI trigger model in SMT. Our experiments focus on the following two issues:

1. How much improvements can we achieve by separately integrating the backward language model and the MI trigger model into our phrase-based SMT system?

2. Can we obtain a further improvement if we jointly apply both models?

5.1 System Overview

Without loss of generality[6], we evaluate our models in a phrase-based SMT system which adapts bracketing transduction grammars to phrasal translation (Xiong et al., 2006). The log-linear model of this system can be formulated as

$$\begin{aligned} w(\mathcal{D}) =& M_T(r^l_{1..n_l}) \cdot M_R(r^m_{1..n_m})^{\lambda_R} \\ & \cdot P_{fL}(e)^{\lambda_{fL}} \cdot exp(|e|)^{\lambda_w} \end{aligned} \tag{20}$$

where $\mathcal{D}$ denotes a derivation, $r^l_{1..n_l}$ are the BTG lexicon rules which translate source phrases to target phrases, and $r^m_{1..n_m}$ are the merging rules which combine two neighboring blocks into a larger block in a straight or inverted order. The translation model M_T consists of widely used phrase and lexical translation probabilities (Koehn et al., 2003).

[6] We have discussed how to integrate the backward language model and the MI trigger model into the standard phrase-based SMT system (Koehn et al., 2003) in Section 3.2 and 4.2 respectively.

The reordering model M_R predicts the merging order (straight or inverted) by using discriminative contextual features (Xiong et al., 2006). P_{fL} is the standard forward n-gram language model.

If we simultaneously integrate both the backward language model P_{bL} and the MI trigger model MI into the system, the new log-linear model will be formulated as

$$\begin{aligned} w(\mathcal{D}) = & M_T(r_{1..n_l}^l) \cdot M_R(r_{1..n_m}^m)^{\lambda_R} \cdot P_{fL}(e)^{\lambda_{fL}} \\ & \cdot P_{bL}(e)^{\lambda_{bL}} \cdot MI(e)^{\lambda_{MI}} \cdot exp(|e|)^{\lambda_w} \end{aligned} \quad (21)$$

5.2 Experimental Setup

Our training corpora[7] consist of 96.9M Chinese words and 109.5M English words in 3.8M sentence pairs. We used all corpora to train our translation model and smaller corpora without the United Nations corpus to build a maximum entropy based reordering model (Xiong et al., 2006).

To train our 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 3.1. Finally, we trained our MI trigger model still on this corpus according to the method in Section 4.1. The trained MI trigger model consists of 2.88M trigger pairs.

We used the NIST MT03 evaluation test data as the development set, and the NIST MT04, MT05 as the test sets. We adopted the case-insensitive BLEU-4 (Papineni et al., 2002) as the evaluation metric, which uses the shortest reference sentence length for the brevity penalty. Statistical significance in BLEU differences is tested by paired bootstrap re-sampling (Koehn, 2004).

5.3 Experimental Results

The experimental results on the two NIST test sets are shown in Table 2. When we combine the backward language model with the forward language model, we obtain 0.49 and 0.56 BLEU points over the baseline on the MT-04 and MT-05 test set respectively. Both improvements are statistically significant ($p < 0.01$). The MI trigger model also achieves statistically significant improvements of 0.33 and 0.44 BLEU points over the baseline on the MT-04 and MT-05 respectively.

Model	MT-04	MT-05
Forward (Baseline)	35.67	34.41
Forward+Backward	36.16+	34.97+
Forward+MI	36.00+	34.85+
Forward+Backward+MI	36.76+	35.12+

Table 2: BLEU-4 scores (%) on the two test sets for different language models and their combinations. +: better than the baseline ($p < 0.01$).

When we integrate both the backward language model and the MI trigger model into our system, we obtain improvements of 1.09 and 0.71 BLEU points over the single forward language model on the MT-04 and MT-05 respectively. These improvements are larger than those achieved by using only one model (the backward language model or the MI trigger model).

6 Analysis

In this section, we will study more details of the two models by looking at the differences that they make on translation hypotheses. These differences will help us gain some insights into how the presented models improve translation quality.

Table 3 shows an example from our test set. The italic words in the hypothesis generated by using the backward language model (F+B) exactly match the reference. However, the italic words in the baseline hypothesis fail to match the reference due to the incorrect position of the word "decree" (法令). We calculate the forward/backward language model score (the logarithm of language model probability) for the italic words in both the baseline and F+B hypothesis according to the trained language models. The difference in the forward language model score is only 1.58, 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 3.52. This larger difference may guarantee that the hypothesis generated by F+B

[7] LDC2004E12, LDC2004T08, LDC2005T10, LDC2003E14, LDC2002E18, LDC2005T06, LDC2003E07 and LDC2004T07.

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
F+B	*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

Table 3: Translation example from the MT-04 test set, comparing the baseline with the backward language model. F+B: forward+backward language model .

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 Force 's trip is remarkable , because it <u>was</u> not an isolated incident .
F+M	Self-Defense Force 's trip is remarkable , because it <u>is</u> not an isolated incident .
Reference	The Self-Defense Forces' trip arouses attention because it <u>is</u> not an isolated incident.

Table 4: Translation example from the 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. F+M: forward language model + MI trigger model.

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.

In Table 4, we present another 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 (F+M) selects the former while the baseline selects the latter. The forward language model score for the baseline hypothesis is -26.41, which is higher than the score of the F+M hypothesis -26.67. 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)[8] is -1.03 while the PMI value of the trigger pair (is, is) is as high as 0.32. Therefore our MI trigger model selects "is" rather than "was".[9] This example illustrates that the MI trigger model is capable of selecting correct words by using long-distance trigger pairs.

7 Conclusion

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 two models have been integrated into the decoder and have shown to improve a state-of-the-art phrase-based SMT system. The first model is the backward language model which uses backward n-grams to predict the current word. We introduced algorithms that directly integrate the backward language model into a CKY-style and a standard phrase-based decoder respectively. The second model is the MI trigger model that incorporates long-distance trigger pairs into language modeling.

Overall improvements are up to 1 BLEU point on the NIST Chinese-to-English translation tasks with large-scale training data. Further study of the two

[8]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.

[9]The overall MI trigger model scores (the logarithm of Eq. (14)) of the baseline hypothesis and the F+M hypothesis are 2.09 and 2.25 respectively.

models indicates that backward n-grams and long-distance triggers provide useful information to improve translation quality.

In future work, we would like to integrate the backward language model into a syntax-based system in a way that is similar to the proposed algorithm shown in Figure 1. We are also interested in exploring more morphologically- or syntactically-informed triggers. For example, a verb in the past tense triggers another verb also in the past tense rather than the present tense.

References

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 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)*, pages 858–867, Prague, Czech Republic, June. Association for Computational Linguistics.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Eugene Charniak, Kevin Knight, and Kenji Yamada. 2003. Syntax-based language models for statistical machine translation. In *Proceedings of MT Summit IX. Intl. Assoc. for Machine Translation*.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Jacques Duchateau, Kris Demuynck, and Patrick Wambacq. 2002. Confidence scoring based on backward language models. In *Proceedings of ICASSP*, pages 221–224, Orlando, FL, April.

Ahmad Emami, Kishore Papineni, and Jeffrey Sorensen. 2007. Large-scale distributed language modeling. In *Proceedings of ICASSP*, pages 37–40, Honolulu, HI, April.

Andrew Finch and Eiichiro Sumita. 2009. Bidirectional phrase-based statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1124–1132, Singapore, August. Association for Computational Linguistics.

Joshua T. Goodman. 2001. A bit of progress in language modeling extended version. Technical report, Microsoft Research.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu. 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*, pages 58–54, Edmonton, Canada, May-June.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July.

Arne Mauser, Saša Hasan, and Hermann Ney. 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*, pages 210–218, Singapore, August. Association for Computational Linguistics.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July.

Matt Post and Daniel Gildea. 2008. Parsers as language models for statistical machine translation. In *Proceedings of AMTA*.

Sylvain Raybaud, Caroline Lavecchia, David Langlois, and Kamel Smaïli. 2009. New confidence measures for statistical machine translation. In *Proceedings of the International Conference on Agents and Artificial Intelligence*, pages 61–68, Porto, Portugal, January.

Roni Rosenfeld, Jaime Carbonell, and Alexander Rudnicky. 1994. Adaptive statistical language modeling: A maximum entropy approach. Technical report, Carnegie Mellon University.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-08: HLT*, pages 577–585, Columbus, Ohio, June. Association for Computational Linguistics.

Andreas Stolcke. 2002. Srilm–an extensible language modeling toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing*, pages 901–904, Denver, Colorado, USA, September.

David Talbot and Miles Osborne. 2007. Randomised language modelling for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 512–519,

Prague, Czech Republic, June. Association for Computational Linguistics.

Dekai Wu. 1996. A polynomial-time algorithm for statistical machine translation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 152–158, Santa Cruz, California, USA, June.

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

Deyi Xiong, Qun Liu, and Shouxun Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 521–528, Sydney, Australia, July. Association for Computational Linguistics.

Ying Zhang, Almut Silja Hildebrand, and Stephan Vogel. 2006. Distributed language modeling for n-best list re-ranking. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 216–223, Sydney, Australia, July. Association for Computational Linguistics.

GuoDong Zhou. 2004. Modeling of long distance context dependency. In *Proceedings of Coling*, pages 92–98, Geneva, Switzerland, Aug 23–Aug 27. COLING.