Source Language Adaptation Approaches for Resource-Poor Machine Translation

Pidong Wang*
Machine Zone, Inc.

Preslav Nakov**
Qatar Computing Research Institute,
HBKU

Hwee Tou Ng†
National University of Singapore

Most of the world languages are resource-poor for statistical machine translation; still, many of them are actually related to some resource-rich language. Thus, we propose three novel, language-independent approaches to source language adaptation for resource-poor statistical machine translation. Specifically, we build improved statistical machine translation models from a resource-poor language POOR into a target language TGT by adapting *and using a large bitext for a related resource-rich language RICH and the same target language TGT. We assume a small POOR–TGT bitext from which we learn word-level and phrase-level paraphrases and cross-lingual morphological variants between the resource-rich and the resource-poor language. Our work is of importance for resource-poor machine translation because it can provide a useful guideline for people building machine translation systems for resource-poor languages.*

Our experiments for Indonesian/Malay–English translation show that using the large adapted resource-rich bitext yields 7.26 BLEU points of improvement over the unadapted one and 3.09 BLEU points over the original small bitext. Moreover, combining the small POOR–TGT bitext with the adapted bitext outperforms the corresponding combinations with the unadapted bitext by 1.93–3.25 BLEU points. We also demonstrate the applicability of our approaches to other languages and domains.

1. Introduction

Contemporary statistical machine translation (SMT) systems learn how to translate from large sentence-aligned bilingual corpora of human-generated translations, called

* 2225 East Bayshore Road, Suite 200, Palo Alto, CA 94303. E-mail: pwang@machinezone.com. The work reported in this article was part of the first author's Ph.D. thesis research in the Department of Computer Science, National University of Singapore.
** Tornado Tower, floor 10, P.O. 5825, Doha, Qatar. E-mail: pnakov@qf.org.qa.
† 13 Computing Drive, Singapore 117417. E-mail: nght@comp.nus.edu.sg.

Submission received: 23 May 2015; revised version received: 10 January 2016; accepted for publication: 15 February 2016.

doi:10.1162/COLI_a_00248

bitexts. Unfortunately, collecting sufficiently large, high-quality bitexts is difficult, and thus most of the 6,500+ world languages are resource-poor for SMT. Fortunately, many of these resource-poor languages are related to some resource-rich language, with whom they overlap in vocabulary and share cognates, which offers opportunities for bitext reuse.

Example pairs of such resource rich–poor languages include Spanish–Catalan, Finnish–Estonian, Swedish–Norwegian, Russian–Ukrainian, Irish–Gaelic Scottish, Standard German–Swiss German, Modern Standard Arabic–Dialectical Arabic (e.g., Gulf, Egyptian), Turkish–Azerbaijani, and so on.

Previous work has already demonstrated the benefits of using a bitext for a related resource-rich language to X (e.g., X = English) to improve machine translation from a resource-poor language to X (Nakov and Ng 2009, 2012). Here we take a different, orthogonal approach: We *adapt* the resource-rich language to get closer to the resource-poor one.

We assume two bitexts: (1) a small bitext for a resource-poor source language S_1 and some target language T, and (2) a large bitext for a related resource-rich source language S_2 and the same target language T. We use these bitexts to learn word-level and phrase-level paraphrases and cross-lingual morphological variants between the resource-poor and resource-rich languages, S_1 and S_2. We propose three approaches to adapt (the source side of) the large bitext for S_2–T: word-level paraphrasing, phrase-level paraphrasing, and text rewriting using a specialized decoder. The first two approaches were proposed in our previous work (Wang, Nakov, and Ng 2012), and the third approach is novel and outperforms the other two in our experiments.

Training on the adapted large bitext S_2'–T yields very significant improvements in translation quality compared with both training on the unadapted large bitext S_2–T, and training on the small bitext for the resource-poor language S_1–T. We further achieve very sizable improvements when combining the small bitext S_1–T with the large adapted bitext S_2'–T, compared with combining the former with the unadapted bitext S_2–T.

Although here we focus on adapting Malay to look like Indonesian, we also demonstrate the applicability of our approach to another language pair, Bulgarian–Macedonian, which is also from a different domain.

The remainder of this article is organized as follows. Section 2 presents an overview of related work. Section 3 introduces our target resource rich–poor language pair: Malay–Indonesian. Then, Section 4 presents our three approaches for source language adaptation. Section 5 describes the experimental set-up, after which we present the experimental results and discussions in Section 6. Section 7 contains deeper analysis of the obtained results. Finally, Section 8 concludes and points to possible directions for future work.

2. Related Work

One relevant line of research is on machine translation between closely related languages, which is arguably simpler than general SMT, and thus can be handled using word-for-word translation, manual language-specific rules that take care of the necessary morphological and syntactic transformations, or character-level translation/transliteration. This has been tried for a number of language pairs including Czech–Slovak (Hajič, Hric, and Kuboň 2000), Turkish–Crimean Tatar (Altintas and Cicekli 2002), Irish–Scottish Gaelic (Scannell 2006), and Macedonian–Bulgarian (Nakov and Tiedemann 2012). In contrast, we have a different objective: We do not carry out full

translation but rather adaptation (since our ultimate goal is to translate into a third language *X*).

A special case of this same line of research is the translation between dialects of the same language, for example, between Cantonese and Mandarin (Zhang 1998), or between a dialect of a language and a standard version of that language, for example, between some Arabic dialect (e.g., Egyptian) and Modern Standard Arabic (Bakr, Shaalan, and Ziedan 2008; Sawaf 2010; Salloum and Habash 2011; Sajjad, Darwish, and Belinkov 2013). Here again, manual rules and/or language-specific tools and resources are typically used. In the case of Arabic dialects, a further complication arises due to the informal status of the dialects, which are not standardized and not used in formal contexts but rather only in informal online media such as social networks, chats, forums, Twitter, and SMS messages, though the Egyptian Wikipedia is one notable exception. This causes further mismatch in domain and genre. Thus, translating from Arabic dialects to Modern Standard Arabic requires, among other things, normalizing informal text to a formal form. Sajjad, Darwish, and Belinkov (2013) first normalized a dialectal Egyptian Arabic to look like Modern Standard Arabic, and then translated the transformed text to English.

In fact, this is a more general problem, which arises with informal sources such as SMS messages and Tweets for just any language (Aw et al. 2006; Han and Baldwin 2011; Wang and Ng 2013; Bojja, Nedunchezhian, and Wang 2015). Here the main focus is on coping with spelling errors, abbreviations, and slang, which are typically addressed using string edit distance, while also taking pronunciation into account. This is different from our task, where we try to adapt good, formal text from one language to another.

A second relevant line of research is on language adaptation and normalization, when done specifically for improving SMT into another language. For example, Marujo et al. (2011) described a rule-based system for adapting Brazilian Portuguese (BP) to European Portuguese (EP), which they used to adapt BP–English bitexts to EP–English. They report small improvements in BLEU for EP–English translation when training on the adapted "EP"–English bitext compared with using the unadapted BP–English (38.55 vs. 38.29 BLEU points), or when an EP–English bitext is used in addition to the adapted/unadapted one (41.07 vs. 40.91 BLEU points). Unlike that work, which heavily relied on language-specific rules, our approach is statistical, and largely language-independent; moreover, our improvements are much more sizable.

A third relevant line of research is on reusing bitexts between related languages without or with very little adaptation, which works well for very closely related languages. For example, our previous work (Nakov and Ng 2009, 2012) experimented with various techniques for combining a small bitext for a resource-poor language (Indonesian or Spanish) with a much larger bitext for a related resource-rich language (Malay or Portuguese), pretending that Spanish is resource-poor; the target language of all bitexts was English. However, that work did not attempt language adaptation, except for very simple transliteration for Portuguese–Spanish that ignored context entirely; because it does not substitute a word with a completely different word, transliteration did not help much for Malay–Indonesian, which use unified spelling. Still, once we have language-adapted the large bitext, it makes sense to try to combine it further with the small bitext; thus, in the following we will directly compare and combine these two approaches.

One alternative, which we do not explore in this work, is to use cascaded translation using a pivot language (Cohn and Lapata 2007; Utiyama and Isahara 2007; Wu and Wang 2009). Unfortunately, using the resource-rich language as a pivot (poor→rich→*X*) would require an additional parallel poor–rich bitext, which we do not have. Pivoting

over the target *X* (rich→*X*→poor) for the purpose of language adaptation, on the other hand, would miss the opportunity to exploit the relationship between the resource-poor and the resource-rich language; this would also be circular since the first step would ask an SMT system to translate its own training data (we only have one rich–*X* bitext).

Yet another alternative approach for improving resource-poor MT is to mine translation bitexts from comparable corpora (Munteanu, Fraser, and Marcu 2004; Snover, Dorr, and Schwartz 2008). This is orthogonal to our efforts here, as our focus is on adapting resources for a related resource-rich language, rather than directly mining source–target translation pairs from comparable corpora.

3. Malay and Indonesian

Malay and Indonesian are closely related, mutually intelligible Austronesian languages with 180 million speakers combined. They have a unified spelling, with occasional differences, for example, *kerana* vs. *karena* ('because'), *Inggeris* vs. *Inggris* ('English'), and *wang* vs. *uang* ('money').

They differ more substantially in vocabulary, mostly because of loan words, where Malay typically follows the English pronunciation, whereas Indonesian tends to follow Dutch, for example, *televisyen* vs. *televisi*, *Julai* vs. *Juli*, and *Jordan* vs. *Yordania*.

Although there are many cognates between the two languages, there are also many false friends, for example, *polisi* means *policy* in Malay but *police* in Indonesian. There are also many partial cognates, for example, *nanti* means both *will* (future tense marker) and *later* in Malay but only *later* in Indonesian.

Thus, fluent Malay and fluent Indonesian can differ substantially. Consider, for example, Article 1 of the *Universal Declaration of Human Rights*:[1]

- *Semua manusia dilahirkan bebas dan samarata dari segi kemuliaan dan hak-hak. Mereka mempunyai pemikiran dan perasaan hati dan hendaklah bertindak di antara satu sama lain dengan semangat persaudaraan.* (Malay)
- *Semua orang dilahirkan merdeka dan mempunyai martabat dan hak-hak yang sama. Mereka dikaruniai akal dan hati nurani dan hendaknya bergaul satu sama lain dalam semangat persaudaraan.* (Indonesian)
- *All human beings are born free and equal in dignity and rights. They are endowed with reason and conscience and should act towards one another in a spirit of brotherhood.* (English)

There is only 50% overlap at the word level, but the actual vocabulary overlap is much higher—for example, there is only one word in the Malay text that does not exist in Indonesian: *samarata* ('equal'). Other differences are due to the use of different morphological forms, for example, *hendaklah* vs. *hendaknya* ('conscience'), derivational variants of *hendak* ('want').

To quantify the similarity between some pairs of languages, we calculated the cosine similarity between them based on the *Universal Declaration of Human Rights*.[2] The results are shown in Table 1. We can see that the average similarity between English

1 http://www.un.org/en/documents/udhr/index.shtml
2 http://www.ohchr.org/EN/UDHR/Pages/SearchByLang.aspx

Table 1
Cosine similarities between some pairs of languages, calculated on the *Universal Declaration of Human Rights* using tokens and filtering out punctuation symbols.

Language Pairs	Cosine Similarity
Malay–Indonesian	0.802
Portuguese–Spanish	0.475
Bulgarian–Macedonian	0.302
French–English	0.033
Spanish–English	0.031
Indonesian–English	0.002
Malay–English	0.001

and {Indonesian, Malay, French, Spanish} is 0.001–0.033, whereas for closely related language pairs it ranges from 0.302 to 0.802. Of course, this cosine calculation compares surface word overlap only and does not take minor morphological variants into consideration. Yet, this gives an idea of the relative proximity between the languages.

Of course, word choice in translation is often a matter of taste. Thus, we asked a native speaker of Indonesian to adapt the Malay version to Indonesian while preserving as many words as possible, and we obtained the following result:

- *Semua manusia dilahirkan bebas dan mempunyai martabat dan hak-hak yang sama. Mereka mempunyai pemikiran dan perasaan dan hendaklah bergaul satu sama lain dalam semangat persaudaraan.* (Indonesian)

Obtaining this latter version from the original Malay text requires three kinds of word-level operations:

- deletion of *dari*, *segi*, and *hati*
- insertion of *yang* and *sama*
- substitution of *samarata* with *mempunyai*, *kemuliaan* with *martabat*, and *dengan* with *dalam*

Additionally, it requires a phrase-level substitution of *bertindak di antara* with *bergaul*.

Unfortunately, we do not have parallel Malay–Indonesian text, which complicates the process of learning when to apply these operations. Thus, in the following we focus our attention on the simplest and most common operation of word/phrase substitution only, leaving the other two operations for future work. There are other potentially useful operations—for example, a correct translation for the Malay *samarata* can be obtained by splitting it into the Indonesian sequence *sama rata*.

Note that simple word substitution is enough in many cases—for example, it is all that is needed for the following Malay–Indonesian sentence pair:

- *KDNK Malaysia dijangka cecah 8 peratus pada tahun 2010.* (Malay)
- *PDB Malaysia akan mencapai 8 persen pada tahun 2010.* (Indonesian)
- *Malaysia's GDP is expected to reach 8 percent in 2010.* (English)

4. Methods

Assuming a resource-rich bitext (Malay–English) and a resource-poor bitext (Indonesian–English), we improve statistical machine translation from the resource-poor language (Indonesian) to English by *adapting* the bitext for the related resource-rich language (Malay) and English to the resource-poor language (Indonesian) and English. We propose three bitext adaptation approaches: word-level paraphrasing, phrase-level paraphrasing, and text rewriting with a specialized decoder.

Given a Malay sentence in the resource-rich Malay–English bitext, we use one of these three adaptation approaches to generate a ranked list of n corresponding adapted "Indonesian" sentences. Then, we pair each such adapted "Indonesian" sentence with the English counterpart in the Malay–English bitext for the Malay sentence it was derived from, thus obtaining a synthetic "Indonesian"–English bitext. Finally, we combine this synthetic bitext with the resource-poor Indonesian–English bitext to train the final Indonesian–English SMT system, using various bitext combination methods.

In the remainder of this section, we first present the word-level paraphrasing approach, followed by the phrase-level paraphrasing approach; then, we describe the text rewriting decoder. Finally, we describe the bitext combination methods we experiment with.

4.1 Word-Level Paraphrasing

Given a Malay sentence, we generate a confusion network containing multiple Indonesian word-level paraphrase options for each Malay word. Each such Indonesian option is associated with a corresponding weight in the network, which is defined as the probability of this option being a translation of the original Malay word, calculated using Equation (1). We decode this confusion network using a large Indonesian language model, thus generating a ranked list of n corresponding adapted "Indonesian" sentences.

In the following we first describe how we generate the word-level Indonesian options and the corresponding weights for the Malay words. Then, we explain how we build, decode, and improve the confusion network.

4.1.1 Inducing Word-Level Paraphrases. We use pivoting over English to induce potential Indonesian word translations for a given Malay word.

First, we build separate directed word alignments for the Malay–English bitext and for the Indonesian–English bitext using IBM model 4 (Brown et al. 1993), and then we combine them using the intersect+grow heuristic (Och and Ney 2003). We then induce Malay–Indonesian word translation pairs assuming that if an Indonesian word i and a Malay word m are aligned to the same English word e, they could be mutual translations. Each translation pair is associated with a conditional probability, estimated by pivoting over English:

$$\Pr(i|m) = \sum_{e} \Pr(i|e)\Pr(e|m) \tag{1}$$

$\Pr(i|e)$ and $\Pr(e|m)$ are estimated using maximum likelihood from the word alignments. Following Callison-Burch, Koehn, and Osborne (2006), we further assume that i is conditionally independent of m given e.

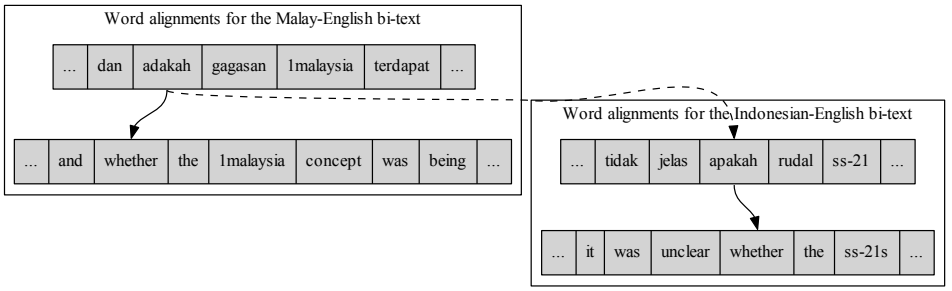

Figure 1
An example of word-level paraphrase induction by pivoting over English. The Malay word *adakah* is aligned with the English word *whether* in the Malay–English bitext (shown with solid arcs). The Indonesian word *apakah* is aligned with the same English word *whether* in the Indonesian–English bitext. We consider *apakah* as a potential translation option of *adakah* (the dashed arc). The other word alignments are not shown.

For example, Figure 1 shows an example that induces an Indonesian word *apakah* as a translation option for the Malay word *adakah*, since the two words are both aligned to the same English word *whether* in the word alignments for the Indonesian–English bitext and the Malay–English bitext, respectively.

4.1.2 Confusion Network Construction. Given a Malay sentence, we construct an Indonesian confusion network, where each Malay word is augmented with a set of alternatives, represented as network transitions: possible Indonesian word translations. The weight of such a transition is the conditional Indonesian–Malay translation probability as calculated by Equation (1); the original Malay word is assigned a weight of 1.

Note that we paraphrase *each* word in the input Malay sentence as opposed to only those Malay words that we believe not to exist in Indonesian (e.g., because they do not appear in our Indonesian monolingual text). This is necessary because of the large number of false friends and partial cognates between Malay and Indonesian (see Section 3).

Finally, we decode the confusion network for a Malay sentence using a large Indonesian language model, and we extract an n-best list. For balance, in case of fewer than n adaptations for a Malay sentence, we randomly repeat some of the available ones. Table 2 shows the 10-best adapted "Indonesian" sentences we generated for the confusion network in Figure 2. According to a native Indonesian speaker, options 1 and 3 in the table are perfect adaptations, options 2 and 5 have a wrong word order, and the rest are grammatical though not perfect.

4.1.3 Further Refinements. Many of our Malay–Indonesian paraphrases are bad: Some have very low probabilities, and others involve rare words for which the probability estimates are unreliable. Moreover, the options we propose for a Malay word are inherently restricted to the small Indonesian vocabulary of the Indonesian–English bitext. We now describe how we address these issues.

Score-based filtering. We filter out translation pairs whose probabilities (Equation (1)) are lower than some threshold (tuned on the development data set), for example, 0.01.

Table 2
The 10-best "Indonesian" sentences extracted from the confusion network in Figure 2.

Rank	"Indonesian" Sentence			
1	pdb	malaysia	akan	mencapai 8 persen pada tahun 2010 .
2	pdb	malaysia	untuk	mencapai 8 persen pada tahun 2010 .
3	pdb	malaysia	diperkirakan	mencapai 8 persen pada tahun 2010 .
4	maka	malaysia	akan	mencapai 8 persen pada tahun 2010 .
5	maka	malaysia	untuk	mencapai 8 persen pada tahun 2010 .
6	pdb	malaysia	dapat	mencapai 8 persen pada tahun 2010 .
7	maka	malaysia	diperkirakan	mencapai 8 persen pada tahun 2010 .
8	sebesar	malaysia	akan	mencapai 8 persen pada tahun 2010 .
9	pdb	malaysia	diharapkan	mencapai 8 persen pada tahun 2010 .
10	pdb	malaysia	ini	mencapai 8 persen pada tahun 2010 .

Improved estimations for $\Pr(i|e)$**.** We concatenate k copies of the small Indonesian–English bitext and one copy of the large Malay–English bitext, where the value of k is selected so that we have roughly the same number of Indonesian and Malay sentences. Then, we generate word-level alignments for the resulting bitext. Finally, we truncate these alignments keeping them for one copy of the original Indonesian–English bitext only. Thus, we end up with improved word alignments for the Indonesian–English bitext, and ultimately with better estimations for Equation (1). Because Malay and Indonesian share many cognates, this improves word alignments for Indonesian words that occur rarely in the small Indonesian–English bitext, but are relatively frequent in the larger Malay–English one; it also helps for some frequent words.

Cross-lingual morphological variants. We increase the Indonesian options for a Malay word using morphology. Because the set of Indonesian options for a Malay word

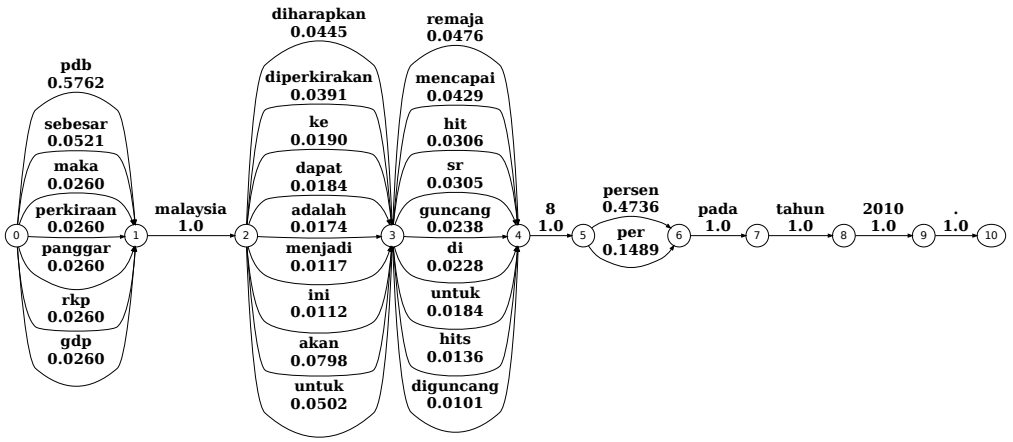

Figure 2
Indonesian confusion network for the Malay sentence *KDNK Malaysia dijangka cecah 8 peratus pada tahun 2010.* Arcs with scores below 0.01 are omitted, and words that exist in Indonesian are not paraphrased (for better readability).

in pivoting is restricted to the Indonesian vocabulary of the small Indonesian–English bitext, this is a severe limitation of pivoting. Thus, assuming a large monolingual Indonesian text, we first build a lexicon of the words in the text. Then, we lemmatize these words using two different lemmatizers: the Malay lemmatizer of Baldwin and Awab (2006), and a similar Indonesian lemmatizer. These two analyzers have different strengths and weaknesses, therefore we combine their outputs to increase recall. Next, we group all Indonesian words that share the same lemma, for example, for *minum* we obtain {*diminum, diminumkan, diminumnya, makan-minum, makananminuman, meminum, meminumkan, meminumnya, meminum-minuman, minum, minum-minum, minum-minuman, minuman, minumanku, minumannya, peminum, peminumnya, perminum, terminum*}. Because Malay and Indonesian are subject to the same morphological processes and share many lemmata, we use such groups to propose Indonesian translation options for a Malay word. We first lemmatize the target Malay word, and then we find all groups of Indonesian words the Malay lemma belongs to. The union of these groups is the set of morphological variants that we will add to the confusion network as additional options for the Malay word. Although the different morphological forms typically have different meanings, for example, *minum* ('drink') vs. *peminum* ('drinker'), in some cases the forms could have the same translation in English, for example, *minum* ('drink', verb) vs. *minuman* ('drink', noun). This is our motivation for trying morphological variants, even though they are almost exclusively derivational, and thus generally quite risky as translational variants. For example, given *seperminuman* ('drinking') in the Malay input, we first find its lemma *minum*, and then we get the above example set of Indonesian words, which contains some reasonable substitutes such as *minuman* ('drink').

We give each Malay–Indonesian morphological variant pair a score Score(i, m), which is one minus the minimum edit distance ratio (Ristad and Yianilos 1998) between the Malay word m and the Indonesian word i:

$$\text{Score}(i, m) = 1 - \frac{\text{EditDistance}(i, m)}{\max(len(i), len(m))} \tag{2}$$

where EditDistance(i, m) is the Levenshtein edit distance between the Indonesian word i and the Malay word m. $len(w)$ is the length of a word w (i.e., the number of characters in w). In the confusion network, the weight of the original Malay word is set to 1. The weight of a morphological option is Score(i, m) multiplied by the highest probability for all pivoting variants for the Malay word—that is, we trust pivoting options more than morphological options. As an example, assuming a morphological variant with Score(i, m) of 0.9 and another pivoting option with a score of 0.8 (Equation (1)), we would finally give the morphological one a weight of $0.9 \times 0.8 = 0.72$ and the pivoting option a weight of 0.8.

4.2 Phrase-Level Paraphrasing

Word-level paraphrasing ignores context when generating Indonesian variants, relying only on the Indonesian language model to make the right contextual choice. This might not be strong enough. Thus, we also try to model context more directly by generating adaptation options at the *phrase level*.

4.2.1 Inducing Phrase-Level Paraphrases. We use standard phrase-based SMT techniques (Koehn et al. 2007) to build separate phrase tables for the Indonesian–English and the Malay–English bitexts. We then pivot over the English phrases to generate Indonesian–Malay phrase pairs. As in the case of word-level pivoting, we derive the paraphrase probabilities from the corresponding probabilities in the two phrase tables, again using Equation (1).

We then use the Moses phrase-based SMT decoder (Koehn et al. 2007) to "translate" the Malay side of the Malay–English bitext to get closer to Indonesian. We use monotone translation, that is, we allow no phrase reordering. We tune the parameters of the log-linear model on a development set using minimum error rate training (MERT) (Och 2003).

4.2.2 Cross-Lingual Morphological Variants. Although phrase-level paraphrasing models context better, it remains limited in the size of its Indonesian vocabulary by the small Indonesian–English bitext, just like word-level paraphrasing. We address this by transforming the Indonesian sentences in the *development* and the *test* Indonesian–English bitexts into confusion networks (Dyer 2007; Du, Jiang, and Way 2010), where we add Malay morphological variants for the Indonesian words, weighting them based on Equation (2). Note that we do not alter the *training* bitext; we just transform the source side of the *development* and the *test* data sets into confusion networks.

4.3 Text Rewriting with a Specialized Decoder

In this section, we introduce a third approach to source language adaptation, which uses a text rewriting decoder to iteratively find the best adaptation for an input sentence.

We first discuss the differences between traditional left-to-right decoders and the text rewriting decoder we propose. We then introduce the decoding algorithm, the different hypothesis producers, and the feature functions we use for source language adaptation.

4.3.1 Differences from Typical Beam-Search Decoders. Beam-search decoders are widely used in natural language processing applications such as SMT, for example, in the phrase-based Moses decoder (Koehn et al. 2007), and automatic speech recognition (ASR), for example, in the HTK hidden Markov model toolkit (Young et al. 2002). Given an input sentence in the source language, various hypotheses about the output sentence in the target language are generated in a left-to-right fashion.

Figure 3 shows an example search tree for the input sentence $s_1 s_2 s_3$, given the following translation options: $\{(s_1, t_2), (s_1 s_2, t_2 t_5), (s_2 s_3, t_6), (s_3, t_4)\}$, where s_i and t_j are source and target words, respectively. Starting from the initial hypothesis, each hypothesis is expanded by adding one more target phrase to the output sentence. This requires keeping a map of which words were translated so far, as the figure shows. Hypotheses with the same maps and the same target output are recombined, and those with the same number of translated words are kept in the same beam. For efficiency reasons, beams are limited in size, and thus only the highest scoring hypotheses make it in each beam. Note that all hypotheses before the last level in the search tree are incomplete, which means that sentence-level feature functions could not be computed exactly for them, for example, type/token ratio (Hardmeier et al. 2013) feature function that models readability.

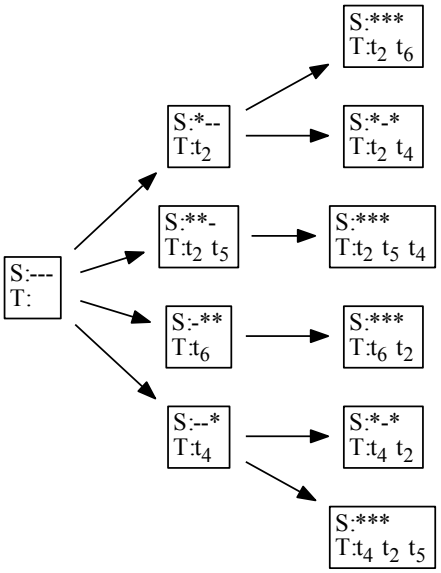

Figure 3
An example search tree of a phrase-based SMT decoder. A source word (in S) that has already been translated is marked with an asterisk (*). The target sentence is shown in T.

Because the decoders used in SMT and ASR typically work at the phrase- or the word-level, they cannot make use of sentence-level features. In contrast, our text rewriting decoder works at the sentence-level, that is, all hypotheses are complete sentences. This means that we can use truly sentence-level features. We will show an example in Section 7.5.

Figure 4 shows the search tree of our decoder for the same input sentence and the same translation options as in the beam decoder example from Figure 3. The search starts from the initial hypothesis, which is then expanded by replacing a source phrase with a target phrase using one phrase pair from the translation options; then, the process continues recursively with each of the new hypotheses.

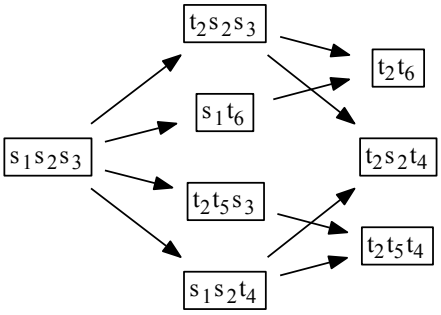

Figure 4
An example search tree of our text rewriting decoder. Each hypothesis is a complete sentence.

4.3.2 Beam-Search Algorithm for Text Rewriting. Given an input sentence, our decoder searches for the best rewriting. It repeats two steps for a number of iterations:

- producing new sentence-level hypotheses from the hypotheses in the current beams, which is carried out by **hypothesis producers**
- scoring these new hypotheses to retain in the beams only the best ones, which is done using **feature functions**

Algorithm 1 describes the search process, which uses lazy pruning, only retaining in the beams the *n*-best hypotheses (as also implemented in the Moses decoder). Hypotheses with the same number of modifications are grouped in the same beam. The maximum number of iterations is equal to the number of tokens in the input sentence, that is, we suppose each token needs at most one modification on average. Upon completion, we select the best hypothesis across all beams.

Algorithm 1 Beam-Search Text Rewriting

INPUT: an **INPUT** sentence of length **N**
RETURN: the best rewritten form for **INPUT**

```
1: initialize hypothesisBeams[0...N] and hypothesisProducers;
2: add the initial hypothesis INPUT to beam hypothesisBeams[0];
3: for i ← 0 to N-1 do
4:    for each hypo in hypothesisBeams[i] do
5:       for each producer in hypothesisProducers do
6:          for each newHypo produced by producer from hypo do
7:             add newHypo to hypothesisBeams[i+1];
8:             prune hypothesisBeams[i+1];
9: return the best hypothesis in hypothesisBeams[0...N];
```

4.3.3 Hypothesis Producers. Hypothesis producers generate new hypotheses by modifying existing ones. We use three types of hypothesis producers:

- *Word-level mapping:* This hypothesis producer uses the word-level pivoted Malay–Indonesian dictionary described in Section 4.1.1. For example, given the hypothesis *KDNK Malaysia dijangka cecah 8.1 peratus pada tahun 2010.*, if the dictionary has the translation pair (*peratus*, *persen*), the following hypothesis will be produced: *KDNK Malaysia dijangka cecah 8.1 persen pada tahun 2010.*
- *Phrase-level mapping:* This hypothesis producer uses the pivoted phrase table described in Section 4.2.1. For example, if the pivoted phrase table contains the phrase pair (*dijangka cecah*, *akan mencapai*), given the hypothesis *KDNK Malaysia dijangka cecah 8.1 peratus pada tahun 2010.*, the new hypothesis *KDNK Malaysia akan mencapai 8.1 peratus pada tahun 2010.* will be generated.
- *Cross-lingual morphological mapping:* This hypothesis producer uses the cross-lingual morphological variants dictionary from a Malay word

to its Indonesian morphological variants described in Section 4.1.3. For example, given the hypothesis *dan untuk meringkaskan pengalamannya?*, if the dictionary contains the morphological variant pair (*meringkaskan*, *meringkas*), the following hypothesis will be produced: *dan untuk meringkas pengalamannya?*

The hypothesis producers presented here are all based on statistical methods. In principle, we can also use some rule-based hypothesis producers to adapt Malay to Indonesian. For example, the number format of Malay is different from that of Indonesian: Malay numbers are written in accordance with the British convention, that is, "." is the decimal point and "," denotes digit grouping, whereas in Indonesian, the roles of "." and "," are switched. Thus, we can build a rule-based hypothesis producer to convert Malay numbers to Indonesian ones, for example, which would convert the hypothesis *KDNK Malaysia dijangka cecah 8.1 peratus pada tahun 2010.* to *KDNK Malaysia dijangka cecah 8,1 peratus pada tahun 2010.* However, such a rule-based hypothesis producer would be language-specific. In the present work, we have chosen to stick to statistical hypothesis producers only in order to keep our decoder as language-independent as possible. This makes it potentially applicable to many closely related language pairs, which we will demonstrate in Section 7.4.

4.3.4 Feature Functions. The text rewriting decoder assesses the quality of a hypothesis based on a log-linear model and a number of feature functions, which can be grouped into two general types.

The first type includes the *count feature functions,* which count the total number of modifications that a given hypothesis producer has made. They allow the decoder to distinguish good hypothesis producers from bad ones. More precisely, if the decoder finds a specific hypothesis producer more useful than others, it can give it a higher weight in order to let it perform more modifications.

The second type includes *general feature functions* such as:

- Indonesian language model score of the adapted "Indonesian" sentence
- Word penalty, that is, the number of tokens in the hypothesis
- Malay word penalty, that is, the number of Malay words in the hypothesis, which are identified using bigram counts from the Indonesian language model: A word w in a hypothesis $\ldots w_{-1}ww_1 \ldots$ is considered a Malay word if both bigrams $w_{-1}w$ and ww_1 do not occur in the Indonesian language model; note that it would be difficult to implement this feature function in a phrase-based SMT decoder such as Moses (Koehn et al. 2007) since hypotheses in Moses only contain incomplete sentences before the last stack, and this feature function asks to see a future word w_1 that has not been generated yet for the last word w; of course, it could also be implemented for words up to w_{-1}, that is, ignoring the last word in the hypothesis, but this would make the implementation different from what is done for LM, and it would also require special treatment of the case of a full hypothesis compared with how partial hypotheses are handled; and the implementation would become even trickier if we want to use higher order n-grams instead of bigrams.

- Word-level mappings: the summation of the logarithms of all conditional probabilities (see Equation (1)) used so far
- Phrase-level mappings: We have four feature functions, each of which is the summation of the logarithms of one of the four probabilities in the pivoted phrase table, that is, forward/reverse phrase translation probability and forward/reverse lexical weighting probability
- Cross-lingual morphological mapping, that is, the summation of the logarithms of all morphological variant mapping scores (see Equation (2)) used so far

4.3.5 Model. We use a log-linear model, which combines all features to obtain the score for a hypothesis h as follows:

$$score(h) = \sum_i \lambda_i f_i(h) \tag{3}$$

where f_i is the ith feature function with weight λ_i.

The text rewriting decoder prunes bad hypotheses based on $score(h)$; it also selects the best hypothesis as the one with the highest $score(h)$ across all beams.

We tune the weights of the feature functions on a development set using pairwise ranking optimization or PRO (Hopkins and May 2011). We optimize BLEU+1 (Liang et al. 2006), a sentence-level approximation of BLEU, as is standard with PRO.

4.4 Combining Bitexts

We have presented our source language adaptation approaches in Sections 4.1, 4.2, and 4.3. Now we explain how we combine the Indonesian–English bitext with the synthetic "Indonesian"–English bitext we have generated. We consider the following three bitext combination approaches:

Simple concatenation. Assuming the two bitexts are of comparable quality, we simply train an SMT system on their concatenation.

Balanced concatenation with repetitions. The two bitexts are not directly comparable. For one thing, "Indonesian"–English is obtained from n-best lists, that is, it has exactly n very similar variants for each Malay sentence. Moreover, the original Malay–English bitext is much larger than the Indonesian–English one and now it has further expanded n times to become "Indonesian"–English, which means it will heavily dominate the concatenation. To counter balance this, we repeat the smaller Indonesian–English bitext enough times to make its number of sentences roughly the same as for "Indonesian"–English; then we concatenate them and train an SMT system on the resulting bitext.

Sophisticated phrase table combination. Finally, we experiment with a method for combining phrase tables proposed in Nakov and Ng (2009, 2012). The first phrase table is extracted from word alignments for the balanced concatenation with repetitions, which are then truncated so that they are kept for only one copy of the Indonesian–English bitext. The second table is built from the simple concatenation. The two tables are then merged as follows: All phrase pairs from the first one are retained, and to them are added those phrase pairs from the second one that are not present in the first one. Each phrase pair retains its original scores, which are further augmented with 1–3 extra

feature scores indicating its origin: The first/second/third feature is 1 if the pair came from the first/second/both table(s), and 0 otherwise. We experiment using all three, the first two, or the first feature only; we also try setting the features to 0.5 instead of 0. This makes six combinations (0, 00, 000, .5, .5.5, .5.5.5); on testing, we use the one that achieves the highest BLEU score on the development set.

Other possibilities for combining the phrase tables include using alternative decoding paths (Birch, Osborne, and Koehn 2007), simple linear interpolation, and direct merging with extra features (Callison-Burch, Koehn, and Osborne 2006); they were previously found inferior to the last two approaches above (Nakov and Ng 2009, 2012).

5. Experiments

With a small Indonesian–English bitext and a larger Malay–English bitext, we use three approaches for source language adaptation to adapt the Malay side of the Malay–English bitext to look like Indonesian, thus obtaining a synthetic "Indonesian"–English bitext. With the synthetic bitext, we run two kinds of experiments:

- *isolated*, where we train an SMT system on the synthetic "Indonesian"–English bitext only
- *combined*, where we combine the synthetic bitext with the original Indonesian–English bitext

In all experiments, we use the same Indonesian–English development set for tuning, and the same Indonesian–English test set for evaluation; see below.

5.1 Data Sets

In our experiments, we use the following data sets, which are required for Indonesian–English SMT:

- **Indonesian–English training bitext (*IN2EN*):** 28,383 sentence pairs; 915,192 English tokens; 796,787 Indonesian tokens
- **Indonesian–English dev bitext (*IN2EN-dev*):** 2,000 sentence pairs; 37,101 English tokens; 35,509 Indonesian tokens
- **Indonesian–English test bitext (*IN2EN-test*):** 2,018 sentence pairs; 36,584 English tokens; 35,708 Indonesian tokens
- **Monolingual English text (*EN-LM*):** 174,443 sentences; 5,071,988 English tokens

Note that the monolingual sentences of *EN-LM* were all collected in the same manner and from the same domains as the other three bilingual texts, in order to reduce the impact of domain mismatch.

We also use a Malay–English set, to be adapted to "Indonesian"–English, and monolingual Indonesian text for building an Indonesian language model:

- **Malay–English training bitext (*ML2EN*):** 290,000 sentence pairs; 8,638,780 English tokens; 8,061,729 Malay tokens
- **Monolingual Indonesian text (*IN-LM*):** 1,132,082 sentences; 20,452,064 Indonesian tokens

We use two bitexts (*IN2EN* and *ML2EN*) to induce word-level and phrase-level paraphrases as described in Sections 4.1.1 and 4.2.1, respectively. Moreover, in Section 4.1.3, we use a large monolingual Indonesian corpus, *IN-LM*, in order to induce Indonesian morphological variants for a Malay word. We built all these monolingual and bilingual data sets from texts we crawled from the Internet.

We further needed a *Malay–Indonesian* development bitext in order to tune the phrase-based SMT decoder in the phrase-level paraphrasing approach of Section 4.2.1, and our source language adaptation decoder of Section 4.3. We created this bitext synthetically: We translated the English side of the *IN2EN-dev* into Malay using Google Translate,[3] and we paired this translated Malay with the Indonesian side of *IN2EN-dev*:

- **Synthetic Malay–Indonesian dev bitext (*ML2IN-dev*):** 2,000 sentence pairs; 34,261 Malay tokens; 35,509 Indonesian tokens

5.2 Baseline Systems

We built five baseline systems – two using a single bitext, *ML2EN* or *IN2EN*, and three combining *ML2EN* and *IN2EN*, using simple concatenation, balanced concatenation, and sophisticated phrase table combination. The last combination is a very strong baseline and the most relevant one that we need to improve upon.

We built each SMT system as follows. Given a training bitext, we built directed word alignments using IBM model 4 (Brown et al. 1993) for both directions, and we combined them using the intersect+grow heuristic (Och and Ney 2003). Based on these alignments, we extracted phrase translation pairs of length up to seven, and we scored them to build a phrase table, where each phrase pair has five features (Koehn 2013): forward and reverse translation probabilities, forward and reverse lexicalized phrase translation probabilities, and a phrase penalty. We further used a 5-gram language model trained using the SRILM toolkit (Stolcke 2002) with modified Kneser-Ney smoothing (Kneser and Ney 1995). We combined all features in a log-linear model, namely: (1) the five features in the phrase table, (2) a language model score, (3) a word penalty, that is, the number of words in the output translation, and (4) distance-based reordering cost.

We tuned the weights of these features by optimizing BLEU (Papineni et al. 2002) on the development set *IN2EN-dev* using MERT (Och 2003), and we used them for translation with the phrase-based SMT decoder of Moses.

We evaluated all systems on the same test set, *IN2EN-test*.

3 http://translate.google.com/.

5.3 Isolated Experiments

In the isolated experiments, we train the SMT system on the adapted "Indonesian"–English bitext only, which allows for a direct comparison to using *ML2EN* or *IN2EN* only.

5.3.1 Using Word-Level Paraphrases. In our word-level paraphrasing experiments, we adapted Malay to Indonesian using three kinds of confusion networks (CN) (see Section 4.1.3 for details):

- ***CN:word*** – using word-level pivoting only
- ***CN:word′*** – using word-level pivoting, with probabilities from word alignments for *IN2EN* that were improved using *ML2EN*
- ***CN:word′+morph*** – *CN:word′* further augmented with cross-lingual morphological variants

There are two parameters to tune on *IN2EN-dev* for the above confusion networks: (1) the minimum pivoting probability threshold for the Malay–Indonesian word-level paraphrases, and (2) the number of n-best Indonesian-adapted sentences that are to be generated for each input Malay sentence. We try $\{0.001, 0.005, 0.01, 0.05\}$ for the threshold and $\{1, 5, 10\}$ for n.

5.3.2 Using Phrase-Level Paraphrases. In our phrase-level paraphrasing experiments, we used pivoted phrase tables (PPT) with the following features for each phrase table entry (in addition to the phrase penalty; see Section 4.2 for more details):

- ***PPT:phrase1*** – only using the forward conditional translation probability
- ***PPT:phrase4*** – using all four conditional probabilities
- ***PPT:phrase4::CN:morph*** – *PPT:phrase4* with a cross-lingual morphological confusion network for the dev/test Indonesian sentences

Here we tune one parameter only: the number of n-best Indonesian-adapted sentences to be generated for each input Malay sentence; we try $\{1, 5, 10\}$. We tune the phrase-level paraphrasing systems on *ML2IN-dev*.

5.3.3 Using a Text Rewriting Decoder. For our text rewriting decoder (DD), we conducted four experiments with different hypothesis producers (see Section 4.3.3 for more details):

- ***DD:word′*** – using only one hypothesis producer, word-level mapping, whose dictionary contains word-level pivoting with probabilities from word alignments for *IN2EN* that were improved using *ML2EN*
- ***DD:word′+morph*** – adding one more hypothesis producer, a cross-lingual morphological mapping hypothesis producer, which uses a dictionary of cross-lingual morphological variants
- ***DD:phrase4*** – only using one phrase-level mapping hypothesis producer, which uses the same pivoted phrase table as *PPT:phrase4*

- ***DD:phrase4+morph*** – this is *DD:phrase4* with a cross-lingual morphological mapping hypothesis producer as for ***DD:word′+morph***

For the first two (word-based) experiments, we tuned the two parameters in Section 5.3.1 on *IN2EN-dev*. For the last two (phrase-based) experiments, we only needed to tune the second parameter of the two. We tried the same values for the two parameters. We tuned the log-linear model of the text rewriting decoder on *ML2IN-dev*.

We also tried to use the word-level and phrase-level hypothesis producers, but this performed about the same as the phrase-level mapping hypothesis producer alone. This may be because the two mappings are extracted from the word alignments of the same Malay–English and Indonesian–English bitexts by pivoting. Thus, we can expect that the phrase-level mapping already contains most, if not all, of the word-level mapping.

5.4 Combined Experiments

These experiments assess the impact of our source language adapted bitext when combined with the original Indonesian–English bitext *IN2EN*, as opposed to combining *ML2EN* with *IN2EN* as was in the last three baselines above. We experimented with the same three combinations: (1) simple concatenation, (2) balanced concatenation, and (3) sophisticated phrase table combination. We tuned the parameters as before; for the last combination, we further had to include in the tuning the extra phrase table features (see Section 4.4 for details).

6. Results and Discussion

In this section, we present the results of our experiments. In all tables, statistically significant improvements ($p < 0.01$), according to Collins, Koehn, and Kučerová's (2005) sign test, over the baseline are in **bold**; in case of two baselines, we use underline for the second baseline.

6.1 Baseline Experiments

The results for the baseline systems are shown in Table 3. We can see that training on *ML2EN* instead of *IN2EN* yields over 4 points absolute drop in BLEU (Papineni

Table 3
The five baselines. The subscript indicates the parameters found on *IN2EN-dev* and used for *IN2EN-test*. The scores that are statistically significantly better than *ML2EN* and *IN2EN* ($p < 0.01$, Collins' sign test) are shown in **bold** and are underlined, respectively.

System	BLEU
ML2EN	14.50
IN2EN	18.67
Simple concatenation	**18.49**
Balanced concatenation	**19.79**
Sophisticated phrase table combination	**20.10**$_{(.5.5)}$

et al. 2002) score, even though *ML2EN* is about 10 times larger than *IN2EN* and both bitexts are from the same domain. This confirms the existence of important differences between Malay and Indonesian. Simple concatenation does not help, but balanced concatenation with repetitions improves by 1.12 BLEU points over *IN2EN*, which shows the importance of giving *IN2EN* a proper weight in the combined bitext. This is further reconfirmed by the sophisticated phrase table combination, which yields an additional absolute gain of 0.31 BLEU points.

6.2 Isolated Experiments

Table 4 shows the results for the isolated experiments. We can see that word-level paraphrasing (*CN:**) improves by up to 5.56 and 1.39 BLEU points over the two baselines (both results are statistically significant). Compared with *ML2EN*, *CN:word* yields an absolute improvement of 4.41 BLEU points, *CN:word′* adds another 0.59, and *CN:word′+morph* adds 0.56 more. The scores for TER (v. 0.7.25) (Snover et al. 2006) and METEOR (v. 1.3) (Banerjee and Lavie 2005) are on par with those for BLEU (NIST v. 13).

Table 4 further shows that the optimal parameters for the word-level systems involve a very low probability cut-off, and a high number of *n*-best sentences. This indicates that they are robust to noise, probably because bad source-side phrases are

Table 4
Isolated experiments. The subscript shows the parameters found on *IN2EN-dev* and used for *IN2EN-test*. The superscript shows the absolute test improvement over the *ML2EN* and the *IN2EN* baselines. Scores that are statistically significantly better than *ML2EN* and *IN2EN* ($p < 0.01$, Collins' sign test) are shown in **bold** and are underlined, respectively. The last line shows system combination results using MEMT.

	System	*n*-gram precision						
		1-gr.	2-gr.	3-gr.	4-gr.	BLEU	TER	METEOR
	ML2EN (baseline)	48.34	19.22	9.54	4.98	14.50	67.14	43.28
	IN2EN (baseline)	55.04	23.90	12.87	7.18	18.67	61.99	54.34
	CN:word	54.50	24.41	13.09	7.35	$\mathbf{18.91}^{(+4.41,+0.24)}_{(0.005,10best)}$	61.94	51.07
	CN:word′	55.05	25.09	13.60	7.69	$\underline{\mathbf{19.50}}^{(+5.00,+0.83)}_{(0.001,10best)}$	61.25	51.97
(i)	*CN:word′+morph*	55.97	25.73	14.06	7.99	$\underline{\mathbf{20.06}}^{(+5.56,+1.39)}_{(0.005,10best)}$	60.31	55.65
	PPT:phrase1	55.11	25.04	13.66	7.80	$\underline{\mathbf{19.58}}^{(+5.08,+0.91)}_{(10best)}$	60.92	51.93
	PPT:phrase4	56.64	26.20	14.53	8.40	$\underline{\mathbf{20.63}}^{(+6.13,+1.96)}_{(10best)}$	59.33	54.23
(ii)	*PPT:phrase4::CN:morph*	56.91	26.53	14.76	8.55	$\underline{\mathbf{20.89}}^{(+6.39,+2.22)}_{(10best)}$	59.30	57.19
	DD:word′	56.57	26.15	14.39	8.18	$\underline{\mathbf{20.39}}^{(+5.89,+1.72)}_{(0.01,10best)}$	59.33	56.66
	DD:word′+morph	56.74	26.22	14.41	8.18	$\underline{\mathbf{20.46}}^{(+5.96,+1.79)}_{(0.005,10best)}$	59.50	56.89
	DD:phrase4	57.14	26.49	14.72	8.49	$\underline{\mathbf{20.85}}^{(+6.35,+2.18)}_{(10best)}$	58.79	57.33
(iii)	*DD:phrase4+morph*	57.35	26.71	14.92	8.63	$\underline{\mathbf{21.07}}^{(+6.57,+2.40)}_{(10best)}$	58.55	57.53
	System combination: (i)+(ii)+(iii)	58.46	27.64	15.46	9.07	$\underline{\mathbf{21.76}}^{(+7.26,+3.09)}$	57.26	58.04

unlikely to match the test-time input. Note also the effect of repetitions: Good word choices are shared by many *n*-best sentences, and thus have higher probability.

The gap between *ML2EN* and *IN2EN* for unigram precision could be explained by vocabulary differences between Malay and Indonesian. Compared with *IN2EN*, all *CN:** models have higher 2/3/4-gram precision. However, *CN:word* has lower unigram precision, which could be due to bad word alignments, as the results for *CN:word*′ show.

When morphological variants are further added, the unigram precision improves by almost 1 BLEU point over *CN:word*′. This shows the importance of morphology for overcoming the limitations of the small Indonesian vocabulary of the *IN2EN* bitext.

The second part of Table 4 shows that phrase-level paraphrasing approach (*PPT:**) performs a bit better. This confirms the importance of modeling context for closely related languages like Malay and Indonesian, which are rich in false friends and partial cognates.

We further see that using more scores in the pivoted phrase table is better. Extending the Indonesian vocabulary with cross-lingual morphological variants is still helpful, though not as much as at the word-level.

The third part of Table 4 shows that text rewriting decoder (*DD:**) performs even better: It further increases the improvements up to 6.57 and 2.40 BLEU points absolutely over the two baselines (statistically significant).

Finally, the combination of the output of the best *PPT*, *CN*, and *DD* systems using MEMT (Heafield and Lavie 2010) yields even further gains, which shows that the three approaches are somewhat complementary. The best BLEU score for our isolated experiments is 21.76, which is already better than all five baselines in Table 3, including the three bitext combination baselines, which only achieve up to 20.10.

6.3 Combined Experiments

Table 5 shows the performance of the three bitext combination strategies (see Section 4.4 for details) when applied to combine *IN2EN* with the original *ML2EN* (i), and with various adapted versions of *ML2EN* (ii–iv).

We can see that for the word-level paraphrasing experiments (*CN:**), all combinations except *CN:word* perform significantly better than their corresponding baselines, but the improvements are most sizeable for simple concatenation. Note that whereas there is a difference of 0.31 BLEU points between the balanced concatenation and the sophisticated combination for the original *ML2EN*, they differ little for the adapted versions. This is probably due to the sophisticated combination assuming that the second bitext is worse than the first one, which is not really the case for the adapted versions: As Table 4 shows, they all outperform *IN2EN*.

Overall, phrase-level paraphrasing (*PPT:**) performs a bit better than word-level paraphrasing, and they are both outperformed by the text rewriting decoder (*DD:**). Finally, system combination with MEMT yields even further gains. These results are consistent with those for the isolated experiments.

7. Further Analysis

In this section, we perform a more in-depth analysis of the obtained results.

Table 5
Combined experiments: BLEU. The subscript indicates the parameters found on *IN2EN-dev* and used for *IN2EN-test*. The absolute test improvement over the corresponding baseline (on top of each column) is in superscript. The scores that are statistically significantly better than *ML2EN* ($p < 0.01$, Collins' sign test) are shown in **bold**. The last line shows system combination results using MEMT.

	Combination with	Combining *IN2EN* with an adapted version of *ML2EN*		
		Simple Concatenation	**Balanced Concatenation**	**Sophisticated Combination**
(i)	+ *ML2EN* (unadapted; baseline)	18.49	19.79	$20.10_{(.5.5)}$
	+ *CN:word*	$\mathbf{19.99}^{(+1.50)}_{(0.001,1best)}$	$20.16^{(+0.37)}_{(0.001,10best)}$	$20.32^{(+0.22)}_{(0.01,10best,.5.5)}$
	+ *CN:word′*	$\mathbf{20.03}^{(+1.54)}_{(0.05,1best)}$	$\mathbf{20.80}^{(+1.01)}_{(0.05,10best)}$	$\mathbf{20.55}^{(+0.45)}_{(0.05,10best,.5.5)}$
(ii)	+ *CN:word′+morph*	$\mathbf{20.60}^{(+2.11)}_{(0.01,10best)}$	$\mathbf{21.15}^{(+1.36)}_{(0.01,10best)}$	$\mathbf{21.05}^{(+0.95)}_{(0.01,5best,00)}$
	+ *PPT:phrase1*	$\mathbf{20.61}^{(+2.12)}_{(1best)}$	$\mathbf{20.71}^{(+0.92)}_{(10best)}$	$20.32^{(+0.22)}_{(1best,000)}$
	+ *PPT:phrase4*	$\mathbf{20.75}^{(+2.26)}_{(1best)}$	$\mathbf{21.08}^{(+1.29)}_{(5best)}$	$\mathbf{20.76}^{(+0.66)}_{(10best,.5.5.5)}$
(iii)	+ *PPT:phrase4::CN:morph*	$\mathbf{21.01}^{(+2.52)}_{(1best)}$	$\mathbf{21.31}^{(+1.52)}_{(5best)}$	$\mathbf{20.98}^{(+0.88)}_{(10best,.5)}$
	+ *DD:word′*	$\mathbf{20.67}^{(+2.18)}_{(0.01,5best)}$	$\mathbf{20.75}^{(+0.96)}_{(0.001,10best)}$	$\mathbf{21.16}^{(+1.06)}_{(0.01,10best,.5.5.5)}$
	+ *DD:word′+morph*	$\mathbf{20.78}^{(+2.29)}_{(0.01,1best)}$	$\mathbf{21.25}^{(+1.46)}_{(0.01,5best)}$	$\mathbf{21.41}^{(+1.31)}_{(0.005,10best,.5.5)}$
	+ *DD:phrase4*	$\mathbf{20.91}^{(+2.42)}_{(5best)}$	$\mathbf{21.20}^{(+1.41)}_{(5best)}$	$\mathbf{20.99}^{(+0.89)}_{(10best,000)}$
(iv)	+ *DD:phrase4+morph*	$\mathbf{21.33}^{(+2.84)}_{(5best)}$	$\mathbf{21.42}^{(+1.63)}_{(5best)}$	$\mathbf{21.08}^{(+0.98)}_{(10best,000)}$
	System combination: (i)+(ii)+(iii)+(iv)	$\mathbf{21.74}^{(+3.25)}$	$\mathbf{21.81}^{(+2.02)}$	$\mathbf{22.03}^{(+1.93)}$

7.1 Paraphrasing Non-Indonesian Words Only

In the *CN:** experiments, we paraphrased *each* word in the Malay input. This was motivated by the existence of false friends such as *polisi* and of partial cognates such as *nanti*. However, doing so also risks proposing worse alternatives, for example, changing *beliau* ('he', respectful) to *ia* ('he', casual), which the weights on the confusion network edges and the language model would not always handle properly. Thus, we tried paraphrasing non-Indonesian words only, that is, those not in *IN-LM*. Because *IN-LM* occasionally contains some Malay-specific words, we also tried paraphrasing words that occur at most t times in *IN-LM*. Table 6 shows that this can yield a loss of up to 1 BLEU point for $t = 0; 10$, and a bit less for $t = 20; 40$.

7.2 Manual Evaluation

We asked a native Indonesian speaker who does not speak Malay to judge whether our "Indonesian" adaptations are more understandable to him than the original Malay input for 100 random sentences. We used two extremes: the conservative *CN:word*,t=0 vs. *CN:word′+morph*. Because the latter is noisy, the top three choices were judged for it. Table 7 shows that *CN:word*,t=0 is better/equal to the original 53%/31% of the time. Thus, it is a very good step in the direction of turning Malay into Indonesian.

Table 6
Paraphrasing non-Indonesian words only: those appearing at most t times in *IN-LM*. The subscript indicates the parameters found on *IN2EN-dev* and used for *IN2EN-test*.

System	BLEU
CN:word, $t = 0$	$17.88_{(0.01,5best)}$
CN:word, $t = 10$	$17.88_{(0.05,10best)}$
CN:word, $t = 20$	$18.14_{(0.01,5best)}$
CN:word, $t = 40$	$18.34_{(0.01,5best)}$
CN:word (i.e., paraphrase all)	$18.91_{(0.005,10best)}$

In contrast, *CN:word′+morph* is typically worse than the original; moreover, those at rank 2 are a bit better than those at rank 1; even compared to the best in top 3, the better:worse ratio is 45%:43%. Still, this latter model works better, which means that phrase-based SMT systems are robust to noise and prefer more variety rather than better translations in the training bitext. That is, humans usually like high precision, whereas what the downstream SMT system really needs should be high recall. Note also that the judgments were at the sentence level, although phrases are sub-sentential, that is, there can be many good phrases to be extracted from a "bad" sentence. For example, *CN:word′+morph* adapted *perisian navigasi kereta 3D di pasaran Malaysia menjelang akhir tahun* ('3D car navigation software hits Malaysia by year-end') to the following three versions (changes are underlined):

- *pertama kali mobil 3D di pasar Malaysia pada akhir tahun*
- *lunak navigasi mobil 3D di pasar Malaysia pada akhir tahun*
- *perangkat navigasi mobil 3D di pasar Malaysia pada akhir tahun*

All three converted *manjelang* ('by') to *pada* ('at'), which is not needed, as *manjelang* is also an Indonesian word. Our human translator did not like the first two versions, but liked the last one better, compared to the original Malay sentence. The first two versions did not adapt *perisian* ('software') correctly, but all three successfully adapted *kereta* to *mobil* ('car'), and also *pasaran* to *pasar* ('market'), which would encourage good phrase pairs in the phrase table extracted from the adapted bitext.

Table 7
Human judgments: Malay versus adapted "Indonesian." A subscript shows the ranking of the sentences, and the parameter values are those from Tables 4 and 6.

System	Better	Equal	Worse
CN:word, $t = 0_{(Rank1)}$	53%	31%	16%
CN:word′+morph$_{(Rank1)}$	38%	8%	54%
CN:word′+morph$_{(Rank2)}$	41%	9%	50%
CN:word′+morph$_{(Rank3)}$	32%	11%	57%
CN:word′+morph$_{(Ranks:1-3)}$	45%	12%	43%

7.3 Reversed Adaptation

In all these experiments, we were adapting the Malay sentences to look like Indonesian. Here we try to reverse the direction of adaptation, that is, to adapt Indonesian to Malay: We thus built an Indonesian-to-Malay confusion network for each dev/test Indonesian sentence using word-level paraphrases extracted with the method of Section 4.1.1. We then use the confusion network as an input to a Malay–English SMT system trained on the *ML2EN* data set. We tried two variations of this idea:

- ***lattice*:** Use Indonesian-to-Malay confusion networks directly as input to the *ML2EN* SMT system; that is, tune a log-linear model using confusion networks for the source side of the *IN2EN-dev* data set, and then evaluate the tuned system using confusion networks for the source side of the *IN2EN-test* dataset.
- ***1-best*:** Decode the Indonesian-to-Malay confusion networks for the source side of *IN2EN-dev* and *IN2EN-test* with a Malay language model (trained on 41,842,640 Malay tokens in the same domain as the *ML2EN* data set) to get the 1-best outputs. Then pair each 1-best output with the corresponding English sentence. Finally, get an adapted "Malay"–English development set and an adapted "Malay"–English test set, and use them to tune and evaluate the *ML2EN* SMT system.

Table 8 shows that both variations perform worse than *CN:word*. We believe this is because *lattice* encodes many options, but does not use a Malay language model, and *1-best* uses a Malay language model, but has to commit to 1-best. In contrast, *CN:word* uses both *n*-best outputs and an Indonesian language model. Designing a similar set-up for reversed adaptation is a research direction that we would like to pursue in future work, since the two reversed adaptation approaches have some advantages over the three adaptation approaches proposed in Section 4; for example, the reversed approaches could be more efficient.

7.4 Adapting Bulgarian to Macedonian to Help Macedonian–English Translation

In order to show the applicability of our framework to other closely related languages and other domains, we experimented with Macedonian (*MK*) and Bulgarian (*BG*), using data from a different, non-newswire domain: the OPUS corpus of movie subtitles (Tiedemann 2009). We used data sets of sizes that are comparable to those in the previous Malay–Indonesian experiments: 160K *MK2EN* and 1.5M *BG2EN* sentence pairs (1.2M and 11.5M English words). Because the sentences of movie subtitles were

Table 8
Reversed adaptation: Indonesian to Malay. The subscript indicates the parameters found on *IN2EN-dev* and used for *IN2EN-test*.

System	BLEU
CN:word (Malay→Indonesian)	$18.91_{(0.005,10best)}$
CN:word (Indonesian→Malay) – lattice	$17.22_{(0.05)}$
CN:word (Indonesian→Malay) – 1-best	$17.77_{(0.001)}$

Table 9
Improving Macedonian–English SMT by adapting Bulgarian to Macedonian. The BLEU scores that are significantly better ($p < 0.01$) than *BG2EN* and *MK2EN* are in **bold** and underlined, respectively. The last line shows system combination results using MEMT.

System	BLEU	TER	METEOR
BG2EN (baseline)	24.57	57.64	41.60
MK2EN (baseline)	26.46	54.55	46.15
Balanced concatenation of *MK2EN* with an adapted *BG2EN*			
+ *BG2EN* (unadapted)	**27.33**	54.61	48.16
+ *CN:word′+morph*	**27.97**	54.08	49.65
+ *PPT:phrase4::CN:morph*	**28.38**	53.35	48.21
+ *DD:phrase4+morph*	**28.44**	53.51	50.95
Combining last four	**29.35**	51.83	51.63

short, we used 10K *MK2EN* sentence pairs for tuning and testing (77K and 72K English words), respectively. For language modeling, we used 9.2M Macedonian and 433M English words.

Table 9 shows that all three approaches (*CN:**, *PPT:**, and *DD:**) outperform the balanced concatenation with unadapted *BG2EN*. Moreover, system combination with MEMT improves even further. This indicates that our approach can work for other pairs of closely related languages and even for other domains.

We should note that the improvements here are less sizeable than those for Malay–Indonesian adaptation. This may be because our monolingual Macedonian data set is much smaller than the monolingual Indonesian data set (10M Macedonian vs. 20M Indonesian words). Also, our monolingual Macedonian data set is too noisy, because it contains many optical character recognition errors, typos, concatenated words, and even some Bulgarian text. Moreover, Macedonian and Bulgarian are arguably somewhat more dissimilar than Malay and Indonesian, as can be seen in Table 1.

7.5 Improving the Readability of the Adapted Bitext

Motivated by Hardmeier et al. (2013), we also experimented with two sentence-level features that aim to improve the readability of the source side of the adapted "Indonesian"–English bitext. The two features are type/token ratio (TTR) and word variation index (OVIX) (Stymne et al. 2013). The latter is a reformulation of TTR that is less sensitive to sentence length. The definitions of TTR and OVIX are shown in Equations (4) and (5), respectively, where $Count(tokens)$ is the number of tokens, and $Count(types)$ is the number of word types.

$$\text{TTR} = \frac{Count(tokens)}{Count(types)} \tag{4}$$

$$\text{OVIX} = \frac{log(Count(tokens))}{log(2 - \frac{log(Count(types))}{log(Count(tokens))})} \tag{5}$$

Table 10
Isolated experiments with readability features, TTR and OVIX. The subscript indicates the parameters found on *IN2EN-dev* and used for *IN2EN-test*. The scores that are statistically significantly better than *ML2EN* and *IN2EN* ($p < 0.01$, Collins' sign test) are shown in **bold** and are underlined, respectively.

System	*n*-gram precision				BLEU
	1-gr.	2-gr.	3-gr.	4-gr.	
ML2EN (baseline)	48.34	19.22	9.54	4.98	14.50
IN2EN (baseline)	55.04	23.90	12.87	7.18	18.67
DD:phrase4	57.14	26.49	14.72	8.49	**20.85**$_{(10best)}$
DD:phrase4+morph	57.35	26.71	14.92	8.63	**21.07**$_{(10best)}$
DD:phrase4+ttr	56.75	26.18	14.53	8.38	**20.63**$_{(10best)}$
DD:phrase4+morph+ttr	57.20	26.52	14.75	8.47	**20.86**$_{(10best)}$
DD:phrase4+ovix	57.05	26.30	14.59	8.39	**20.70**$_{(10best)}$
DD:phrase4+morph+ovix	57.12	26.44	14.64	8.39	**20.75**$_{(5best)}$

We added the two features to the best isolated systems (*DD:phrase4* and *DD:phrase4+morph*) in Table 4. The results are shown in Table 10, where we can see that the two features yield slightly lower BLEU scores, which is similar to what Hardmeier et al. (2013) observed. Hardmeier et al. (2013) also found that improving readability may result in a lower BLEU score, as simple texts would likely not match complicated reference translations, especially if the reference translations were not produced with high readability in mind in the first place.

7.6 Our Text Rewriting Decoder vs. Phrase-Level Paraphrasing

The results of our experiments show that phrase-level paraphrasing outperformed word-level paraphrasing, and they were both outperformed by the text rewriting decoder. Here, we discuss the differences between our text rewriting decoder and using phrase-level paraphrasing with a standard SMT phrase-based decoder like Moses:

- The standard phrase-based SMT decoder works at the phrase level, whereas our text rewriting decoder works at the sentence level, which allows it to make use of sentence-level features (e.g., the readability features in Section 7.5).
- Because of the general framework of our text rewriting decoder presented in Section 4.3, it can use a broader type of feature functions (e.g., a Malay word penalty, which would be hard to integrate in an SMT decoder, as discussed in Section 4.3.4).
- Adding the cross-lingual morphological variants to the text rewriting decoder is more straightforward, that is, as a hypothesis producer. In contrast, in the phrase-level paraphrasing approach, we had to transform the sentences in the *development* and the *test* sets into confusion networks, which contain the additional morphological variants. Alternatively, we could have also hacked the phrase tables to include the morphological variants.

- The text rewriting decoder can easily use rule-based hypothesis producers, for example, the number adaptation discussed in Section 4.3.3 can be added to the decoder as a hypothesis producer. It could also be implemented using XML markup in the Moses SMT decoder (Koehn 2013).

Ultimately, the greatest strength of our decoder is its flexibility. It provides access to a wide space of feature functions and hypothesis producers, and allows us to easily test many different ideas. Furthermore, because the original input sentence could be itself a valid hypothesis, the structure of evaluating rewrites is a natural fit to our problem.

8. Conclusion and Future Work

We have presented work on improving machine translation for a resource-poor language by making use of resources for a related resource-rich language. This is an important line of research because most world languages remain resource-poor for machine translation, while many, if not most, of them are actually related to some resource-rich language(s). We have proposed three approaches, which all *adapt* a bitext for a related resource-rich language to get closer to the resource-poor one: (1) word-level paraphrasing using confusion networks, (2) phrase-level paraphrasing using pivoted phrase tables, and (3) adaptation using a specialized text rewriting decoder.

More precisely, assuming a large *RICH–TGT* bitext for a resource-rich language and a small *POOR–TGT* bitext for a related resource-poor language, we use one of the three proposed approaches to adapt the *RICH* side of the *RICH–TGT* bitext to get closer to *POOR*, thus obtaining a synthetic "*POOR*"*–TGT* bitext, which we then combine with the original *POOR–TGT* bitext to improve the translation from *POOR* to *TGT*.

Using a large bitext for the resource-rich Malay–English language pair and a small bitext for the resource-poor Indonesian–English language pair, and adapting the former to look like the latter, we have achieved very significant improvements over several baselines: (1) +7.26 BLEU points over an unadapted version of the Malay–English bitext, (2) +3.09 BLEU points over the Indonesian–English bitext, and (3) 1.93–3.25 BLEU points over three bitext combinations of the Malay–English and Indonesian–English bitexts. We thus have shown the potential of the idea that source-language adaptation of a resource-rich bitext can improve machine translation for a related resource-poor language. Moreover, we have demonstrated the applicability of the general approach to other languages and domains.

The work presented here is of importance for resource-poor machine translation because it can provide a useful guideline for people building statistical machine translation systems for resource-poor languages. They can adapt bitexts for related resource-rich languages to the resource-poor language, and thus subsequently improve the resource-poor language translation using the adapted bitexts.

This work leaves several interesting directions for future research:

- One direction is to add more word editing operations, for example, word deletion, insertion, splitting, and concatenation (because we mainly focused on word substitution in this study).
- Another promising direction is to add more sentence-level feature functions to the text rewriting decoder to further improve language adaptation.

- Future work could also experiment with other phrase table combination methods, for example, Foster and Kuhn (2007) proposed a mixture model whose weights are learned with an EM algorithm (Foster, Chen, and Kuhn 2013).
- Another direction is to add word reordering. In the current work, we assume no word reordering is necessary (apart from what can be achieved within a phrase), but there actually can exist word-order differences between closely related languages.
- A further direction is to utilize the relationships between the source and the target sides of the input resource-rich bitext to perform language adaptation, since only the source side was used in our current work. For example, Malay–Indonesian adaptation may benefit from adapting a Malay word, considering the English words that this Malay word is aligned to in the Malay–English bitext.
- Another direction is to experiment with other closely related language pairs, for example, the language pairs mentioned in Section 1.
- Finally, further work may apply the language adaptation idea to other linguistic problems, for example, adapt the Malay training data for part-of-speech tagging to "Indonesian" in order to help Indonesian part-of-speech tagging.

Acknowledgments

We would like to give special thanks to Christian Hadiwinoto, Harta Wijaya, and Aldrian Obaja Muis, native speakers of Indonesian, for their help in the linguistic analysis of the input and output of our system. We would also like to thank the reviewers for their constructive comments and suggestions, which have helped us improve the quality of this article.

This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office. Some of the results presented in this article were published in Wang, Nakov, and Ng (2012) and in the Ph.D. thesis of the first author (Wang 2013).

References

Altintas, Kemal and Ilyas Cicekli. 2002. A machine translation system between a pair of closely related languages. In *Proceedings of the 17th International Symposium on Computer and Information Sciences*, ISCIS '02, pages 192–196, Orlando, FL.

Aw, AiTi, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for SMS text normalization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, ACL-COLING '06, pages 33–40, Sydney.

Bakr, Hitham Abo, Khaled Shaalan, and Ibrahim Ziedan. 2008. A hybrid approach for converting written Egyptian colloquial dialect into diacritized Arabic. In *Proceedings of the 6th International Conference on Informatics and Systems*, INFOS '08, pages 27–33, Cairo.

Baldwin, Timothy and Su'ad Awab. 2006. Open source corpus analysis tools for Malay. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, LREC '06, pages 2212–2215, Genoa.

Banerjee, Satanjeev and Alon Lavie. 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*, pages 65–72, Ann Arbor, MI.

Birch, Alexandra, Miles Osborne, and Philipp Koehn. 2007. CCG supertags in factored statistical machine translation. In *Proceedings of the Second Workshop on*

Statistical Machine Translation, WMT '07, pages 9–16, Prague.

Bojja, Nikhil, Arun Nedunchezhian, and Pidong Wang. 2015. Machine translation in mobile games: Augmenting social media text normalization with incentivized feedback. In *Proceedings of the 15th Machine Translation Summit (MT Users' Track)*, volume 2, pages 11–16, Miami, FL.

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

Callison-Burch, Chris, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 17–24, New York, NY.

Cohn, Trevor and Mirella Lapata. 2007. Machine translation by triangulation: Making effective use of multi-parallel corpora. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, ACL '07, pages 728–735, Prague.

Collins, Michael, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, ACL '05, pages 531–540, Ann Arbor, MI.

Du, Jinhua, Jie Jiang, and Andy Way. 2010. Facilitating translation using source language paraphrase lattices. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 420–429, Cambridge, MA.

Dyer, Chris. 2007. The University of Maryland translation system for IWSLT 2007. In *Proceedings of the International Workshop on Spoken Language Translation*, IWSLT '07, pages 180–185, Trento.

Foster, George, Boxing Chen, and Roland Kuhn. 2013. Simulating discriminative training for linear mixture adaptation in statistical machine translation. In *Proceedings of the XIV Machine Translation Summit*, pages 183–190, Nice.

Foster, George and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, WMT '07, pages 128–135, Prague.

Hajič, Jan, Jan Hric, and Vladislav Kuboň. 2000. Machine translation of very close languages. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ANLP '00, pages 7–12, Seattle, WA.

Han, Bo and Timothy Baldwin. 2011. Lexical normalization of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, ACL-HLT '11, pages 368–378, Portland, OR.

Hardmeier, Christian, Sara Stymne, Jörg Tiedemann, and Joakim Nivre. 2013. Docent: A document-level decoder for phrase-based statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, ACL '13, pages 193–198, Sofia.

Heafield, Kenneth and Alon Lavie. 2010. Combining machine translation output with open source: The Carnegie Mellon multi-engine machine translation scheme. *The Prague Bulletin of Mathematical Linguistics*, 93(1):27–36.

Hopkins, Mark and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1352–1362, Edinburgh.

Kneser, Reinhard and Hermann Ney. 1995. Improved backing-off for *m*-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '95, pages 181–184, Detroit, MI.

Koehn, Philipp. 2013. Moses user manual and code guide. Paper available at `http://www.statmt.org/moses/manual/manual.pdf`.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, ACL '07, pages 177–180, Prague.

Liang, Percy, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proceedings of the 21st International Conference on*

Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, COLING-ACL '06, pages 761–768, Sydney.

Marujo, Luís, Nuno Grazina, Tiago Luís, Wang Ling, Luísa Coheur, and Isabel Trancoso. 2011. BP2EP - adaptation of Brazilian Portuguese texts to European Portuguese. In *Proceedings of the 15th Conference of the European Association for Machine Translation*, EAMT '11, pages 129–136, Leuven.

Munteanu, Dragos Stefan, Alexander Fraser, and Daniel Marcu. 2004. Improved machine translation performance via parallel sentence extraction from comparable corpora. In *Proceedings of the Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics Annual Meeting*, HLT-NAACL '04, pages 265–272, Boston, MA.

Nakov, Preslav and Hwee Tou Ng. 2009. Improved statistical machine translation for resource-poor languages using related resource-rich languages. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, EMNLP '09, pages 1358–1367, Singapore.

Nakov, Preslav and Hwee Tou Ng. 2012. Improving statistical machine translation for a resource-poor language using related resource-rich languages. *Journal of Artificial Intelligence Research*, 44:179–222.

Nakov, Preslav and Jörg Tiedemann. 2012. Combining word-level and character-level models for machine translation between closely-related languages. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL '12, pages 301–305, Jeju Island.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, ACL '03, pages 160–167, Sapporo.

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

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, ACL '02, pages 311–318, Philadelphia, PA.

Ristad, Eric and Peter Yianilos. 1998. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532.

Sajjad, Hassan, Kareem Darwish, and Yonatan Belinkov. 2013. Translating dialectal Arabic to English. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL '13, pages 1–6, Sofia.

Salloum, Wael and Nizar Habash. 2011. Dialectal to Standard Arabic paraphrasing to improve Arabic–English statistical machine translation. In *Proceedings of the Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties*, pages 10–21, Stroudsburg, PA.

Sawaf, Hassan. 2010. Arabic dialect handling in hybrid machine translation. In *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas*, AMTA '10, Denver, CO.

Scannell, Kevin P. 2006. Machine translation for closely related language pairs. In *Proceedings of the LREC 2006 Workshop on Strategies for Developing Machine Translation for Minority Languages*, pages 103–109, Genoa.

Snover, Matthew, Bonnie Dorr, and Richard Schwartz. 2008. Language and translation model adaptation using comparable corpora. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 857–866, Honolulu, HI.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the Association for Machine Translation in the Americas*, AMTA '06, pages 223–231, Cambridge, MA.

Stolcke, Andreas. 2002. SRILM–an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, ICSLP '02, pages 901–904, Denver, CO.

Stymne, Sara, Jörg Tiedemann, Christian Hardmeier, and Joakim Nivre. 2013. Statistical machine translation with readability constraints. In *Proceedings of the 19th Nordic Conference on Computational Linguistics*, NODALIDA '13, pages 375–386, Oslo.

Tiedemann, Jörg. 2009. News from OPUS—a collection of multilingual parallel corpora with tools and interfaces.

In *Proceedings of the Recent Advances in Natural Language Processing*, RANLP '09, pages 237–248, Borovets.

Utiyama, Masao and Hitoshi Isahara. 2007. A comparison of pivot methods for phrase-based statistical machine translation. In *Proceedings of the Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL-HLT '07, pages 484–491, Rochester, NY.

Wang, Pidong. 2013. *A Text Rewriting Decoder with Application to Machine Translation*. Ph.D. thesis, National University of Singapore, Singapore.

Wang, Pidong, Preslav Nakov, and Hwee Tou Ng. 2012. Source language adaptation for resource-poor machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 286–296, Jeju Island.

Wang, Pidong and Hwee Tou Ng. 2013. A beam-search decoder for normalization of social media text with application to machine translation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '13, pages 471–481, Atlanta, GA.

Wu, Hua and Haifeng Wang. 2009. Revisiting pivot language approach for machine translation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, ACL-IJCNLP '09, pages 154–162, Singapore.

Young, Steve, Gunnar Evermann, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Valtcho Valtchev, and Phil Woodland. 2002. *The HTK book*, volume 3. Cambridge University Engineering Department.

Zhang, Xiaoheng. 1998. Dialect MT: A case study between Cantonese and Mandarin. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, ACL-COLING '98, pages 1460–1464, Quebec.