Syntax-based Rewriting for Simultaneous Machine Translation

He He

Computer Science
University of Maryland
hhe@cs.umd.edu

Alvin Grissom II, Jordan Boyd-Graber

Computer Science
University of Colorado
{Alvin.Grissom,
Jordan.Boyd.Graber}
@colorado.edu

Hal Daumé III

Computer Science and UMIACS
University of Maryland
hal@cs.umd.edu

Abstract

Divergent word order between languages causes delay in simultaneous machine translation. We present a sentence rewriting method that generates more monotonic translations to improve the speed-accuracy tradeoff. We design grammaticality and meaning-preserving syntactic transformation rules that operate on constituent parse trees. We apply the rules to reference translations to make their word order closer to the source language word order. On Japanese-English translation (two languages with substantially different structure), incorporating the rewritten, more monotonic reference translation into a phrase-based machine translation system enables better translations faster than a baseline system that only uses gold reference translations.

1 Introduction

Simultaneous interpretation is challenging because it demands both quality and speed. Conventional **batch** translation waits until the entire sentence is completed before starting to translate. This merely optimizes translation quality and often introduces undesirable lag between the speaker and the audience. Simultaneous interpretation instead requires a tradeoff between quality and speed. A common strategy is to translate independently translatable segments as soon as possible. Various segmentation methods (Fujita et al., 2013; Oda et al., 2014) reduce translation delay; they are limited, however, by the unavoidable word reordering between languages with drastically different word orders. We show an example of Japanese-English translation in Figure 1. Consider the **batch** translation: in English, the verb *change* comes immediately after the subject *We*, whereas in Japanese it comes at the end

of the sentence; therefore, to produce an intelligible English sentence, we must translate the object after the final verb is observed, resulting in one large and painfully delayed segment.

To reduce structural discrepancy, we can apply syntactic transformations to make the word order of one language closer to the other. Consider the **monotone** translation in Figure 1. By passivizing the English sentence, we can cache the subject and begin translating before observing the final verb. Furthermore, by using the English possessive, we mimic the order of the Japanese genitive construction. These transformations enable us to divide the input into shorter segments, thus reducing translation delay.

To produce such monotone translations, a straightforward approach is to incorporate interpretation data into the learning of a machine translation (MT) system, because human interpreters use a variety of strategies (Shimizu et al., 2014; Camayd-Freixas, 2011; Tohyama and Matsubara, 2006) to fine-tune the word order. Shimizu et al. (2013) shows that this approach improves the speed-accuracy tradeoff. However, existing parallel simultaneous interpretation corpora (Shimizu et al., 2014; Matsubara et al., 2002; Bendazzoli and Sandrelli, 2005) are often small, and collecting new data is expensive due to the inherent costs of recording and transcribing speeches (Paulik and Waibel, 2010). In addition, due to the intense time pressure during interpretation, human interpretation has the disadvantage of simpler, less precise diction (Camayd-Freixas, 2011; Al-Khanji et al., 2000) compared to human translations done at the translator's leisure, allowing for more introspection and precise word choice.

We aim to address the data scarcity problem and combine translators' lexical precision and interpreters' syntactic flexibility. We propose to **rewrite the reference translation** in a way that uses the original lexicon, obeys standard grammar rules of

Source:	我々は	政府の	構造 や	組織を	変更 すべき だ
	We-TOP	**government**-GEN	structure and	composition-ACC	**change** should COP

Batch:	我々は ‖	政府の構造や組織を変更すべき だ
	We	should **change** the structure and composition **of the government**

Monotone: 我々は ‖	政府の ‖	構造や組織を ‖	変更すべきだ
the government's	structure and composition	should **be changed** by us

Figure 1: Divergent word order between language pairs can cause long delays in simultaneous translation: Segments (‖) mark the portions of the sentence that can be translated together. (Case markers: topic (TOP), genitive (GEN), accusative (ACC), copula (COP).)

the target language, preserves the original semantics, and yields more monotonic translations. We then train the MT system with the rewritten references so that it learns how to produce low-latency translations from the data. A data-driven approach to learning these rewriting rules is hampered by the dearth of parallel data: we have few examples of text that have been both interpreted and translated. Therefore, we design syntactic transformation rules based on linguistic analysis of the source and the target languages. We apply these rules to parsed text and decide whether to accept the rewritten sentence based on the amount of delay reduction. In this work, we focus on Japanese to English translation, because (i) Japanese and English have significantly different word orders (SOV vs. SVO); and consequently, (ii) the syntactic constituents required earlier by an English sentence often come late in the corresponding Japanese sentence.

We evaluate our approach using standard machine translation data (the Reuters newsfeed Japanese-English corpus) in a simultaneous translation setting. Our experimental results show that including the rewritten references into the learning of a phrase-based MT system results in a better speed-accuracy tradeoff against both the original and the rewritten reference translations.

2 The Problem of Delay Reduction

Simultaneous interpretation has two goals: producing good translations *and* producing them promptly. However, most existing parallel corpora and MT systems do not address the issue of **delay** during translation. We explicitly adapt the training data by rewriting rules to reduce delay. We first define translation delay and describe—in general terms—our rewriting rules. In the next section, we describe the rules in more detail.

While we are motivated by real-time interpretation, to simplify our problem, we assume that we have perfect text input. Given this constraint, a typical simultaneous interpretation system (Sridhar et al., 2013; Fujita et al., 2013; Oda et al., 2014) produces partial translations of consecutive segments in the source sentence and concatenates them to produce a complete translation. We define the *translation delay* of a sentence as the average number of tokens the system has to observe between translation of two consecutive segments (denoted by # words/seg).[1] For instance, the minimum delay of 1 word/seg is achieved when we translate immediately upon hearing a word. At test time, when the input is segmented, the delay is the average segment length. During the data preprocessing step of rewriting, we calculate delay from word alignments (Section 4).

Given a reference batch translation x, we apply a set of rewriting rules $\mathcal{R}$ to x to minimize its delay. A rewriting rule $r \in \mathcal{R}$ is a mapping that takes the constituent parse tree of x as input and outputs a modified parse tree, which specifies a rewritten sentence x'. The tree-editing operation includes node deletion, insertion, and swapping, as well as induced changes of word form and node label. A valid transformation rule should rearrange constituents in x to follow the word order of the input sentence as closely as possible, subject to grammatical constraints and preservation of the original meaning.

---

[1]Ideally, delay should be based on time lapse. However, timestamping is not applicable to typical MT corpus, therefore we approximate it by number of tokens and ignore decoding time.

3   Transformation Rules

We design a variety of syntactic transformation rules for Japanese-English translation motivated by their structural differences. Our rules cover verb, noun, and clause reordering. While we specifically focus on Japanese to English, many rules are broadly applicable to `SOV` to `SVO` languages.

3.1   Verb Phrases

The most significant difference between Japanese and English is that the head of a verb phrase comes at the end of Japanese sentences. In English, it occupies one of the initial positions. We now introduce rules that can postpone a head verb.

Passivization and Activization   In Japanese, the standard structure of a sentence is NP_1 NP_2 `verb`, where case markers following the verb indicate the voice of the sentence. However, in English, we have NP_1 `verb` NP_2, where the form of the verb indicates its voice. Changing the voice is particularly useful when NP_2 (object in an active-voice sentence and subject in a passive-voice sentence) is long. By reversing positions of `verb` and NP_2, we are not held back by the upcoming verb and can start to translate NP_2 immediately. Figure 1 shows an example in which passive voice can help make the target and source word orders more compatible, but it is not the case that passivizing every sentence would be a good idea; sometimes making a passive sentence active makes the word orders more compatible if the objects are relatively short:

O: The talk was **denied** by the boycott group spokesman.
R: The boycott group spokesman **denied** the talk.

Quotative Verbs   *Quotative* verbs are verbs that, syntactically and semantically, resemble *said* and often start an independent clause. Such verbs are frequent, especially in news, and can be moved to the end of a sentence:

O: **They announced** that the president will restructure the division.
R: The president will restructure the division, **they announced**.

In addition to quotative verbs, candidates typically include factive (e.g., *know*, *realize*, *observe*), factive-like (e.g., *announce*, *determine*), belief (e.g., *believe*, *think*, *suspect*), and antifactive (e.g., *doubt*, *deny*) verbs. When these verbs are followed by a clause (`S` or `SBAR`), we move the verb and its subject to the end of the clause.

While some exploratory work automatically extracts factive verbs, to our knowledge, an exhaustive list does not exist. To obtain a list with reasonable coverage, we exploit the fact that Japanese has an unambiguous quotative particle, *to*, that precedes such verbs.[2] We identify all of the verbs in the Kyoto corpus (Neubig, 2011) marked by the quotative particle and translate them into English. We then use these as our quotative verbs.[3]

3.2   Noun Phrases

Another difference between Japanese and English lies in the order of adjectives and the nouns they modify. We identify two situations where we can take advantage of the flexibility of English grammar to favor sentence structures consistent with positions of nouns in Japanese.

Genitive Reordering   In Japanese, genitive constructions always occur in the form of X *no* Y, where Y belongs to X. In English, however, the order may be reversed through the *of* construction. Therefore, we transform constructions NP_1 of NP_2 to possessives using the apostrophe-s, NP_2'(s) NP_1 (Figure 1). We use simple heuristics to decide if such a transformation is valid. For example, when X / Y contains proper nouns (e.g., *the City of New York*), numbers (e.g., *seven pounds of sugar*), or pronouns (e.g., *most of them*), changing them to the possessive case is not legal.

***that* Clause**   In English, clauses are often modified through a pleonastic pronoun. E.g., *It is* `ADJP` *to/that* `SBAR`/`S`. In Japanese, however, the subject (clause) is usually put at the beginning. To be consistent with the Japanese word order, we move the modified clause to the start of the sentence: *To* `S`/`SBAR` *is* `ADJP`. The rewritten English sentence is still grammatical, although its structure is less frequent in common English usage. For example,

O: **It is important** to remain watchful.
R: To remain watchful **is important**.

---

[2] We use a morphological analyzer to distinguish between the conjunction and quotative particles. Examples of words marked by this particle include 見られる (*expect*), 言う (*say*), 思われる (*seem*), する (*assume*), 信じる (*believe*) and so on.
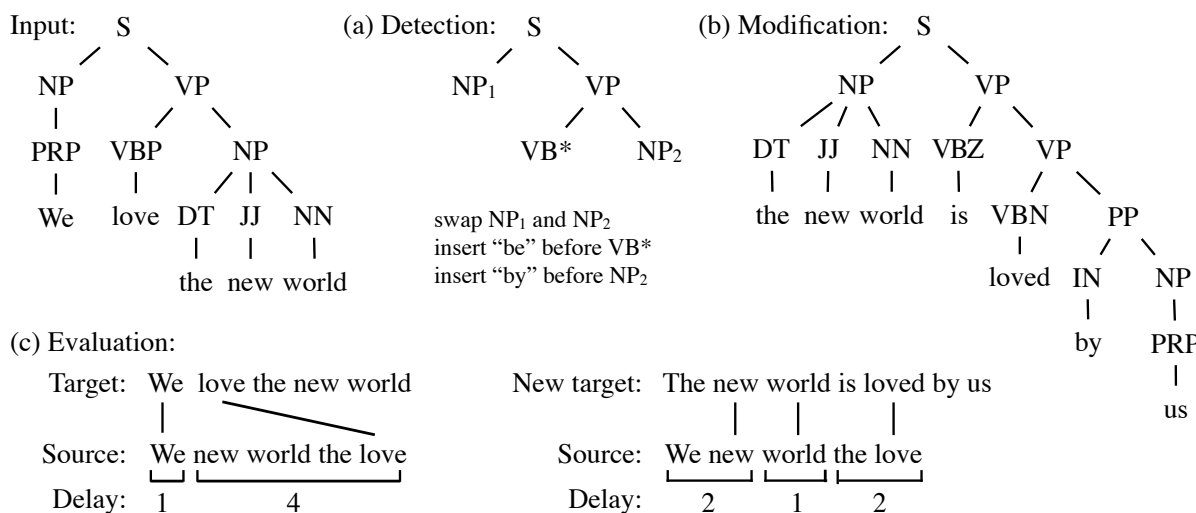
[3] We also include the phrase *It looks like*.

Input: S
NP VP
PRP VBP NP
We love DT JJ NN
the new world

(a) Detection: S
NP₁ VP
VB* NP₂

swap NP₁ and NP₂
insert "be" before VB*
insert "by" before NP₂

(b) Modification: S
NP VP
DT JJ NN VBZ VP
the new world is VBN PP
loved IN NP
by PRP
us

(c) Evaluation:

Target: We love the new world
Source: We new world the love
Delay: 1   4

New target: The new world is loved by us
Source: We new world the love
Delay: 2   1   2

Figure 2: An example of applying the passivization rule to create a translation reference that is more monotonic.

3.3 Conjunction Clause

In Japanese, clausal conjunctions are often marked at the end of the initial clause of a compound sentence. In English, however, the order of clauses is more flexible. We can therefore reduce delay by reordering the English clauses to mirror how they typically appear in Japanese. Below we describe rules reversing the order of clauses connected by these conjunctions:

- Clausal conjunctions: *because (of)*, *in order to*
- Contrastive conjunctions: *despite*, *even though*, *although*
- Conditionals: *(even) if, as a result (of)*
- Misc: *according to*

In standard Japanese, such conjunctions include *no de*, *kara*, *de mo* and so on. The sentence often appears in the form of S_2 `conj`, S_1. In English, however, two common constructions are

S_1 `conj` S_2: We should march **because** winter is coming.

`conj` S_2, S_1: **Because** winter is coming, we should march.

To follow the Japanese clause order, we adapt the above two constructions to

S_2, `conj'` S_1: Winter is coming, **because of this**, we should march.

Here `conj'` represents the original conjunction word appended with simple pronouns/phrases to refer to S_2. For example, *because → because of this, even if → even if this is the case*.

4 Sentence Rewriting Process

We now turn our attention to the implementation of the syntactic transformation rules described above. Applying a transformation consists of three steps:

1. **Detection:** Identify nodes in the parse tree for which the transformation is applicable;
2. **Modification:** Transform nodes and labels;
3. **Evaluation:** Compute delay reduction, and decide whether to accept the rewritten sentence.

Figure 2 illustrates the process using passivization as an example. In the detection step, we find the subtree that satisfies the condition of applying a rule. In this case, we look for an `S` node whose children include an `NP` (denoted by `NP₁`), the subject, and a `VP` to its right, such that the `VP` node has a leaf `VB*`, the main verb,[4] followed by another `NP` (denoted by `NP₂`), the object. We allow the parent nodes (`S` and `VP`) to have additional children besides the matched ones. They are not affected during the transformation. In the modification step, we swap the subject node and object node; we add the verb *be* in its correct form by checking the tense of the verb and the form of `NP₂`;[5] and we add the preposition *by* before the subject. The process is executed recursively throughout the parse tree.

---

[4] The main verb excludes *be* and *have* when it indicates tense (e.g., *have done*).

[5] We use the Nodebox linguistic library (`https://www.nodebox.net/code`) to detect and modify word forms.

Although our rules are designed to minimize long range reordering, there are exceptions.[6] Thus applying a rule does not always reduce delay. In the evaluation step, we compare translation delay before and after applying the rule. We accept a rewritten sentence if its delay is reduced; otherwise, we revert to the input sentence. Since we do not segment sentences during rewriting, we must estimate the delay.

To estimate the delay, we use word alignments. Figure 2c shows the source Japanese sentence in its word-for-word English translation and alignments from the target words to the source words. The first English word, *We*, is aligned to the first Japanese word; it can thus be treated as an independent segment and translated immediately. The second English word, *love*, is aligned to the last Japanese word, which means the system cannot start to translate until four more Japanese words are revealed. This alignment therefore forms a segment with delay of four words/seg. Alignments of the following words come before the source word aligned to *love*; hence, they are already translated in the previous segment and we do not double count their delay. In this example, the delay of the original sentence is 2.5 word/seg; after rewriting, it is reduced to 1.7 word/seg. Therefore, we accept the rewritten sentence. However, when the subject phrase is long and the object phrase is short, a swap may not reduce delay.

We can now formally define the **delay**. Let e_i be the ith target word in the input sentence x and a_i be the maximum index among indices of source words that e_i aligned to. We define the delay of e_i as $d_i = \max(0, a_i - \max_{j<i} a_j)$. The delay of x is then $\sum_{i=1}^{N} d_i / N$, where the sum is over all aligned words except punctuation and stopwords.

Given a set of rules, we need to decide which rules to apply and in what order. Fortunately, our rules have little interaction with each other, and the order of application has a negligible effect. We apply the rules, roughly, sequentially in order of complexity: if the output of current rule is not accepted, the sentence is reverted to the last accepted version.

	Train	Tune	Test
Ja	21.3M	30.2k	23.3k
En-GD	16.8M	23.8k	18.5k
En-RW	16.8M	24.1k	18.7k

Table 1: Number of words in the training, tuning, and test datasets. En-GD and En-RW represent the gold reference set and the rewritten reference set.

5   Experiments

We evaluate our method on the Reuters Japanese-English corpus of news articles (Utiyama and Isahara, 2003). For training the MT system, we also include the EIJIRO dictionary entries and the accompanying example sentences.[7] Statistics of the dataset are shown in Table 1. The rewritten translation is generally slightly longer than the gold translation because our rewriting often involves inserting pronouns (e.g. *it*, *this*) for antecedents.

We use the `TreebankWordTokenizer` from NLTK (Bird et al., 2009) to tokenize English sentences and Kuromoji Japanese morphological analyzer[8] to tokenize Japanese sentences. Our phrase-based MT system is trained by Moses (Koehn et al., 2003) with standard parameters settings. We use GIZA++ (Och and Ney, 2003) for word alignment and k-best batch MIRA (Cherry and Foster, 2012) for tuning. The translation quality is evaluated by BLEU (Papineni et al., 2002) and RIBES (Isozaki et al., 2010).[9] To obtain the parse trees for English sentences, we use the Stanford Parser (Klein and Manning, 2003) and the included English model.

5.1   Quality of Rewritten Translations

After applying the rewriting rules (Section 4), Table 2 shows the percentage of sentences that are candidates and how many rewrites are accepted. The most generalizable rules are passivization and delaying quotative verbs. We rewrite 32.2% of sentences, reducing the delay from 9.9 words/seg to 6.3 words/seg per segment for rewritten sentences and from 7.8 words/seg to 6.7 words/seg overall.

---

[6]For example, in clause transformation, the Japanese conjunction *moshi*, which is clause initial, may appear at the beginning of a sentence to emphasize conditionals, although its appearance is relatively rare.

[7]Available at `http://eijiro.jp`

[8]Available at `http://www.atilika.org/`

[9]In contrast to BLEU, RIBES is an order-sensitive metric commonly used for translation between Japanese and English.

	verb	voice	noun	conj.
Applicable %	39.9	50.0	26.4	4.8
Accepted %	22.5	24.0	51.2	38.4

Table 2: Percentage of sentences that each rule category can be applied to (Applicable) and the percentage of sentences for which the rule results in a more monotonic sentence (Accepted).

We evaluate the quality of our rewritten sentences from two perspectives: grammaticality and preserved semantics. To examine how close the rewritten sentences are to standard English, we train a 5-gram language model using the English data from the Europarl corpus, consisting of 46 million words, and use it to compute perplexity. Rewriting references increases the perplexity under the language model only slightly: from 332.0 to 335.4. To ensure that rewrites leave meaning unchanged, we use the SEMAFOR semantic role labeler (Das et al., 2014) on the original and modified sentence; for each role-labeled token in the reference sentence, we examine its corresponding role in the rewritten sentence and calculate the average accuracy acrosss all sentences. Even ignoring benign lexical changes—for example, *he* becoming *him* in a passivized sentence—95.5% of the words retain their semantic roles in the rewritten sentences.

Although our rules are conservative to minimize corruption, some errors are unavoidable propagation of parser errors. For example, the sentence *the London Stock Exchange closes at 1230 GMT today* is parsed as:[10]

(S (NP *the London Stock Exchange*)
    (VP (VBZ *closes*)
         (PP *at 1230*)
         (NP *GMT today*)))

GMT today is separated from the PP as an NP and is mistaken as the object. The passive version is then *GMT today is closed at 1230 by the London Stock Exchange*. Such errors could be reduced by skipping nodes with low inside/outside scores given by the parser, or skipping low-frequency patterns. However, we leave this for future work.

5.2 Segmentation

At test time, we use right probability (Fujita et al., 2013, RP) to decide when to start translating a

---

[10]For simplicity we show the shallow parse only.

sentence. As we read in the source Japanese sentence, if the input segment matches an entry in the learned phrase table, we query the RP of the Japanese/English phrase pair. A higher RP indicates that the English translation of this Japanese phrase will likely be followed by the translation of the next Japanese phrase. In other words, translation of the two consecutive Japanese phrases is monotonic, thus, we can begin translating immediately. Following (Fujita et al., 2013), if the RP of the current phrase is lower than a fixed threshold, we cache the current phrase and wait for more words from the source sentence; otherwise, we translate all cached phrases. Finally, translations of segments are concatenated to form a complete translation of the input sentence.

5.3 Speed/Accuracy Trade-off

To show the effect of rewritten references, we compare the following MT systems:

- **GD:** only gold reference translations;
- **RW:** only rewritten reference translations;
- **RW+GD:** both gold and the rewritten references; and
- **RW-LM+GD:** using gold reference translations but using the rewritten references for training the LM and for tuning.

For RW+GD and RW-LM+GD, we interpolate the language models of GD and RW. The interpolating weight is tuned with the rewritten sentences. For RW+GD, we combine the translation models (phrase tables and reordering tables) of RW and GD by fill-up combination (Bisazza et al., 2011), where all entries in the tables of RW are preserved and entries from the tables of GD are added if new.

Increasing the RP threshold increases interpretation delay but improves the quality of the translation. We set the RP threshold at 0.0, 0.2, 0.4, 0.8 and finally 1.0 (equivalent to batch translation). Figure 3 shows the BLEU/RIBES scores vs. the number of words per segement as we increase the threshold. Rewritten sentences alone do not significantly improve over the baseline. We suspect this is because the transformation rules sometimes generate ungrammatical sentences due to parsing errors, which impairs learning. However, combining RW and GD results in a better speed-accuracy tradeoff: the RW+GD curve completely dominates other curves in Figure 3a, 3c. Thus, using more monotone translations improves simultaneous machine translation, and because RW-LM+GD is about

(a) BLEU w.r.t. gold ref

(b) RIBES w.r.t. gold ref

(c) BLEU w.r.t. rewritten ref
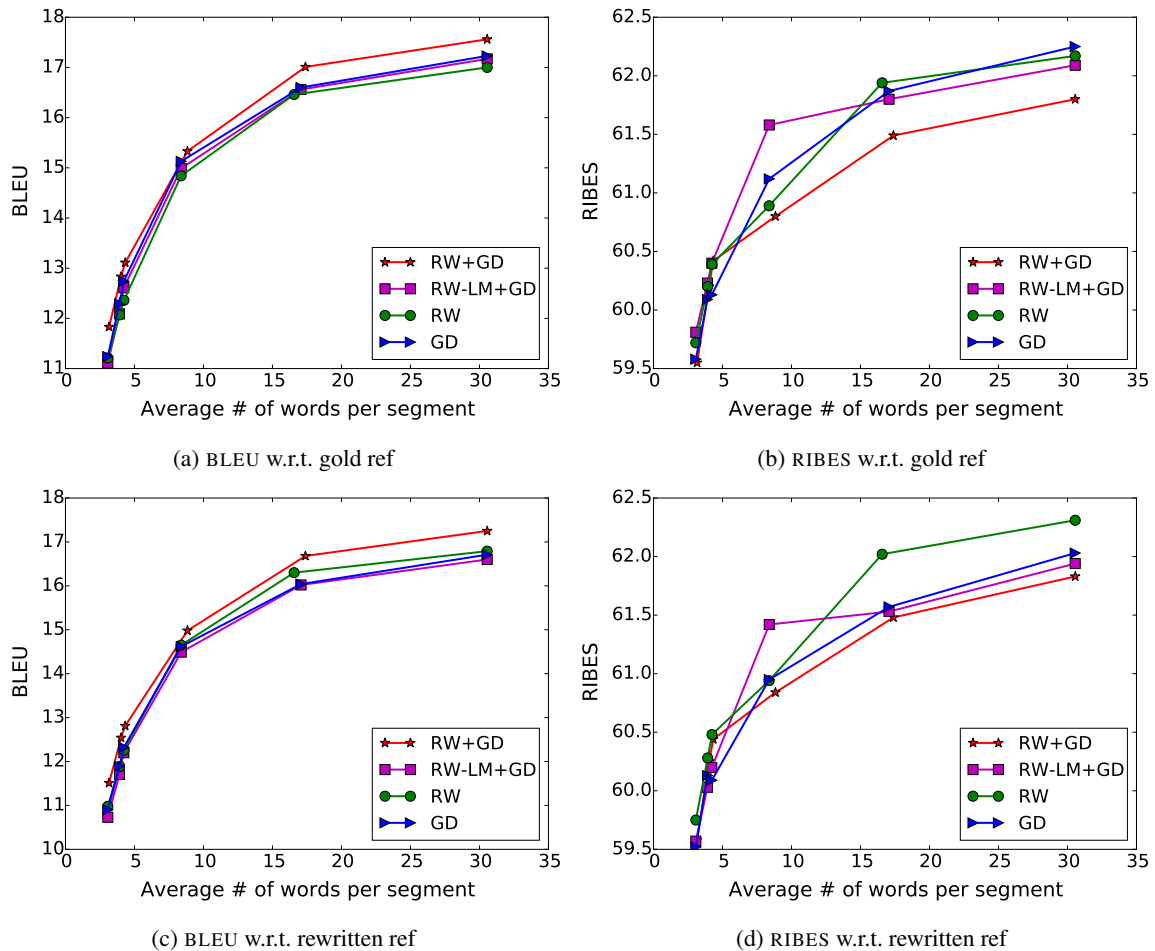

(d) RIBES w.r.t. rewritten ref

Figure 3: Speed/accuracy tradeoff curves: BLEU (left) / RIBES (right) versus translation delay (average number of words per segment).

the same as GD, the major improvement likely comes from the translation model from rewritten sentences.

The right two plots recapitulate the evaluation with the RIBES metric. This result is less clear, as MT systems are optimized for BLEU and RIBES penalizes word reordering, making it difficult to compare systems that intentionally change word order. Nevertheless, RW is comparable to GD on gold references and superior to the baseline on rewritten references.

5.4 Effect on Verbs

Rewriting training data not only creates lower latency simultaneous translations, but it also improves *batch* translation. One reason is that SOV to SVO translation often drops the verb because of long range reordering. (We see this for Japanese here, but this is also true for German.) Similar word orders in the source and target results in less reordering and improves phrase-based MT (Collins

	Translation			
	GD	RW	RW+GD	Gold ref
# of verbs	1971	2050	**2224**	2731

Table 3: Number of verbs in the test set translation produced by different models and the gold reference translation. Boldface indicates the number is significantly larger than others (excluding the gold ref) according to two-sample t-tests with $p < 0.001$.

et al., 2005; Xu et al., 2009). Table 3 shows the number of verbs in the translations of the test sentences produced by GD, RW, RW+GD, as well as the number in the gold reference translation. Both RW and RW+GD produce more verbs (a statistically significant result), although RW+GD captures the most verbs.

Ref	**he also said** that the real dangers for the euro lay in the potential for divergences in the domestic policy needs among the various participating nations of the single currency.
GD	**he** also for the euro, is a real danger to launch a single currency in many different countries and domestic policies on the need for the possibility of a difference.
RW	**he** also for the euro is a real danger to launch a single currency in many different countries and domestic policies to the needs of the possibility of a difference, **he said**.

Table 4: Example of translation produced by GD and RW.

5.5 Error Analysis

Table 4 compares translations by GD and RW. RW correctly puts the verb *said* at the end, while GD drops the final verb. However, RW still produces *he* at the beginning (also the first word in the Japanese source sentence). This is because our current segmentation strategy do not preserve words for *later* translation—a note-taking strategy used by human interpreters.

6 Related Work

Previous approaches to simultaneous machine translation have employed explicit interpretation strategies for coping with delay. Two major approaches are segmentation and prediction.

Most segmentation strategies are based on heuristics, such as pauses in speech (Fügen et al., 2007; Bangalore et al., 2009), comma prediction (Sridhar et al., 2013) and phrase reordering probability (Fujita et al., 2013). Learning-based methods have also been proposed. Oda et al. (2014) find segmentations that maximize the BLEU score of the final concatenated translation by dynamic programming. Grissom II et al. (2014) formulate simultaneous translation as a sequential decision making problem and uses reinforcement learning to decide when to translate. One limitation of these methods is that when learning with standard batch MT corpus, their gain can be restricted by natural word reordering between the source and the target sentences, as explained in Section 1.

In an SOV-SVO context, methods to predict unseen words are proposed to alleviate the above restriction. Matsubara et al. (1999) predict the English verb in the target sentence and integrates it syntactically. Grissom II et al. (2014) predict the final verb in the source sentence and decide when to use the predicted verb with reinforcement learning.

Nevertheless, unless the predictor considers contextual and background information, which human interpreters often rely on for prediction (Hönig, 1997; Camayd-Freixas, 2011), such a prediction task is inherently hard.

Unlike previous approaches to simultaneous translation, we directly adapt the training data and transform a translated sentence to an "interpreted" one. We can, therefore, take advantage of the abundance of parallel batch-translated corpora for training a simultaneous MT system. In addition, as a data preprocessing step, our approach is orthogonal to the others, with which it can be easily combined.

This work is also related to preprocessing reordering approaches (Xu et al., 2009; Collins et al., 2005; Galley and Manning, 2008; Hoshino et al., 2013; Hoshino et al., 2014) in batch MT for language pairs with substantially different word orders. However, our problem is different in several ways. First, while the approaches resemble each other, our motivation is to reduce translation delay. Second, they reorder the source sentence, which is nontrivial and time-consuming when the sentence is incrementally revealed. Third, rewriting the target sentence requires the output to be grammatical (for it to be used as reference translation), which is not a concern when rewriting source sentences.

7 Conclusion

Training MT systems with more monotonic (interpretation-like) sentences improves the speed-accuracy tradeoff for simultaneous machine translation. By designing syntactic transformations and rewriting batch translations into more monotonic translations, we reduce the translation delay. MT systems trained on the rewritten reference translations learn interpretation strategies implicitly from the data.

Our rewrites are based on linguistic knowledge and inspired by techniques used by human interpreters. They cover a wide range of reordering phenomena between Japanese and English, and more generally, between SOV and SVO languages. A natural extension is to automatically extract such rules from parallel corpora. While there exist approaches that extract syntactic tree transformation rules automatically, one of the difficulties is that most parallel corpora is dominated by lexical paraphrasing instead of syntactic paraphrasing.

Acknowledgments

References

Raja Al-Khanji, Said El-Shiyab, and Riyadh Hussein. 2000. On the use of compensatory strategies in simultaneous interpretation. *Journal des Traducteurs*, 45(3):548–577.

Srinivas Bangalore, Vivek Kumar Rangarajan Sridhar, Prakash Kolan, Ladan Golipour, and Aura Jimene. 2009. Real-time incremental speech-to-speech translation of dialogs. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Claudio Bendazzoli and Annalisa Sandrelli. 2005. An approach to corpus-based interpreting studies: Developing EPIC (european parliament interpreting corpus). In *Proceedings of Challenges of Multidimensional Translation*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

Arianna Bisazza, Nick Ruiz, and Marcello Federico. 2011. Fill-up versus interpolation methods for phrase-based SMT adaptation. In *Proceedings of International Workshop on Spoken Language Translation (IWSLT)*.

Erik Camayd-Freixas. 2011. Cognitive theory of simultaneous interpreting and training. In *Proceedings of the 52nd Conference of the American Translators Association*.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*.

Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, , and Noah A. Smith. 2014. Frame-semantic parsing. *Computational Linguistics*, 40(1).

Christian Fügen, Alex Waibel, , and Muntsin Kolss. 2007. Simultaneous translation of lectures and speeches. *Machine Translation*, 21(4):209–252.

Tomoki Fujita, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2013. Simple, lexicalized choice of translation timing for simultaneous speech translation. In *Proceedings of Interspeech*.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

Alvin C. Grissom II, He He, Jordan Boyd-Graber, John Morgan, and Hal Daumé III. 2014. Don't until the final verb wait: Reinforcement learning for simultaneous machine translation. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

Hans G. Hönig. 1997. Using text mappings in teaching consecutive interpreting. *Translation and Translation Theory*, pages 19–34.

Sho Hoshino, Yusuke Miyao, Katsuhito Sudoh, and Masaaki Nagata. 2013. Two-stage pre-ordering for Japanese-to-English statistical machine translation. In *International Joint Conference on Artificial Intelligence (IJCNLP)*.

Sho Hoshino, Hubert Soyer, Yusuke Miyao, and Akiko Aizawa. 2014. Japanese to english machine translation using preordering and compositional distributed semantics. In *Workshop on Asian Translation*.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Brooke Cowan Nicola Bertoldi, Wade Shen, Richard Zens Christine Moran, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbs. 2003. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*.

Shigeki Matsubara, Katsuhiko Toyama, and Yasuyoshi Inagaki. 1999. Sync/trans: Simultaneous machine interpretation between English and Japanese. In *Advanced Topics in Artificial Intelligence*, pages 134–143. Springer.

Shigeki Matsubara, Akira Takagi, Nobuo Kawaguchi, and Yasuyoshi Inagaki. 2002. Bilingual spoken

monologue corpus for simultaneous machine interpretation research. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*.

Graham Neubig. 2011. The Kyoto free translation task. *Available online at http://www. phontron. com/kftt*.

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

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2014. Optimizing segmentation strategies for simultaneous speech translation. In *Proceedings of the Association for Computational Linguistics (ACL)*, June.

Kishore Papineni, Salim Roukos, Todd Ward, , and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*.

Matthias Paulik and Alex Waibel. 2010. Spoken language translation from parallel speech audio: Simultaneous interpretation as slt training data. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Hiroaki Shimizu, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2013. Constructing a speech translation system using simultaneous interpretation data. In *Proceedings of International Workshop on Spoken Language Translation (IWSLT)*.

Hiroaki Shimizu, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2014. Collection of a simultaneous translation corpus for comparative analysis. In *International Language Resources and Evaluation (LREC)*.

Vivek Kumar Rangarajan Sridhar, John Chen, Srinivas Bangalore Andrej Ljolje, and Rathinavelu Chengalvarayan. 2013. Segmentation strategies for streaming speech translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Hitomi Tohyama and Shigeki Matsubara. 2006. Collection of simultaneous interpreting patterns by using bilingual spoken monologue corpus. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*.

Masao Utiyama and Hitoshi Isahara. 2003. Reliable measures for aligning japanese-english news articles and sentences. In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*.

Peng Xu, Jaeho Kang, Michael Ringgaard, , and Franz Och. 2009. Using a dependency parser to improve SMT for subject-object-verb language. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.