System Combination for Machine Translation Based on Text-to-Text Generation

Wei-Yun Ma
Department of Computer Science
Columbia University
New York, NY 10027, USA
ma@cs.columbia.edu

Kathleen McKeown
Department of Computer Science
Columbia University
New York, NY 10027, USA
kathy@cs.columbia.edu

Abstract

In this paper, we introduce a novel translation system combination framework using a text-to-text generation technique. We are motivated by the observation that for many translation sentences, some of their constituents may be poorly translated, while others may be well translated. Moreover, it is often the case that other systems can provide good alternatives to problematic constituents. In our approach, the system constructs paraphrase lattices representing all possible hypotheses for the same source sentence. We then use a text-to-text generator operating on those lattices to generate those hypotheses. We filter ungrammatical combinations using a feature-based lexicalized tree adjoining grammars (FB-LTAG) and then use a TER-based metric to compute a consensus score function to select the best translation among grammatical hypotheses. The system combination gains 1.38 BLEU points over the best individual system.

1 Introduction

Recently many MT combination approaches have been presented. Consensus network (CN) decoding (Matusov et al., 2006; Rosti et al., 2007; He et al. 2008; Rosti et al. 2010; Leusch and Ney, 2010) is one of the most successful approaches, in which the words in all hypotheses are aligned with a backbone hypothesis. A word-based lattice is then formed with word alternatives, including nulls, each with associated scores from voting or other confidence scores. Then, the combined translation sentence(s) can be produced with the same word order as the backbone by selecting the path with the highest score(s) along the lattice.

In this paper, rather than use the CN decoding framework, we borrow the idea of text-to-text generation, which has been used successfully for sentence fusion as part of text summarization (Barzilay and McKeown 2005; Marsi and Krahmer 2005; Fillapova and Strube 2008), to do our translation combination. In text-to-text generation, a sentence fusion is produced from many input sentences by selecting phrases that appear in the majority of input sentences; it is typically used in summarization and questions answering. To maintain the syntactic quality of the fused sentence, syntactic constraints are applied when aligning constituents during the construction of the fusion lattice. Our motivation for using this strategy is based on the observation that for many translation sentences, even though some of their constituents are poorly translated, other constituents are well translated. Furthermore, frequently other systems can provide good alternatives for problematic constituents. We propose replacing those problematic phrases with good translations, while retaining the same overall syntactic structure for the sentence. Table 1 provides an example from our experiment to illustrate the approach.

We use a phrase-based lattice structure to which text to text generation can be applied. Here the definition of "phrase" is word, linguistic constituent or clause. We call the lattice *a paraphrase lattice* because each arc represents paraphrases of a target phrase. In order to maintain a well-formed syntactic structure of the generation result, we require the paraphrases in the lattice to be linguistic constituents with the same phrase type. This syntactic constraint takes into account that even if paraphrases have the same meaning, the substitution of one for another is nonetheless inappropriate when their original syntactic contexts are different.

This results in a well-formed syntactic structure. In addition, to ensure the grammatical correctness of generated hypotheses, for our work on the language pair of Chinese-English, we also propose a grammatical error detector for English based on feature-based lexicalized tree adjoining grammars (FB-LTAG) to filter out ungrammatical generated hypotheses. In our experiment, 14% generated hypotheses are detected to have grammatical errors and thus are filtered out.

Given a pool of syntactically valid generated hypotheses, our goal is to select the best translation among them. We use a translation edit rate (TER)-based metric that measures consensus between the hypothesis and all system outputs to select the best translation among valid generated hypotheses.

2 An Example of Generated Hypothesis

Source Sentence: 阿富汗塔利班组织绑架二十三名南韩人质,并**设定今天日落为最后期限,要求南韩自阿国撤军**,否则将杀害人质,
Reference Sentence: The Taliban in Afghanistan have kidnapped 23 South Korean hostages and **fixed sunset today as the deadline, demanding that South Korea withdraw its troops from Afghanistan** or they would kill the hostages .
A system translation Afghanistan 's Taliban group abducted 23 South Korean hostages , and **set the deadline today , calling for the withdrawal of troops from South Korea** , otherwise they will kill the hostages .
One generated hypothesis Afghanistan 's Taliban group abducted 23 South Korean hostages , and **set the deadline today , calling for South Korea to withdraw its troops from Afghanistan** , otherwise they will kill the hostages .

Table 1. An example of generated hypothesis. The generated hypothesis is the system translation except that the entire VP (bold part) of that system translation is replaced with another, better, VP provided by another system (not shown here).

3 System Overview

Our text-to-text generation procedure for the translation combination involves the following steps:

1. Collect the hypotheses from multiple systems for an input source sentence. In this work, the source-to-target word alignments are available from the individual systems. For a given sentence, every system's translation is labeled as one backbone.

2. Use a syntactic parser to parse all backbones.

3. For all linguistic constituents of each backbone, collect their corresponding paraphrases from other backbones. Note that every single word is also regarded as a linguistic constituent/phrase in this work. Using Fig 1 as the example, "instead of rice" is the paraphrase of "other than rice".

4. Construct a paraphrase lattice for each backbone in the form of a target-to-target phrase table based on the collected paraphrases, such as Fig 2, the paraphrase lattice for the sys1 translation in Fig1.

5. Decode those paraphrase lattices using a standard decoder to generate all possible target language realizations.

6. Filter out ungrammatical generated hypotheses using a grammatical error detector. We then place every backbone's valid generated hypotheses in a single hypothesis pool.

7. From the pool, select the best translation using a scoring function. In this work, we tried a TER-based score metric, as well as TER in combination with a language model.

In the following sections, we describe steps 3-7.

4 Paraphrase Extraction

Given a backbone linguistic phrase, *b*, our aim is to extract every valid paraphrase-pair *(b,h)*, in which *h* is a linguistic paraphrase of b, drawn from other backbones. There are several conditions for *(b,h)*:

- Because of the consideration of efficiency based on the amount of generated hypotheses, we limit the maximum word length of *b* and *h* to be 15 words.
- *b* and *h* individually satisfy the standard definition of bilingual word alignments (Och and Ney 2004); that is for a *b* or *h*, there exists a source phrase *f* such that the phrase pair *(b,f)* or *(h,f)* is consistent with the word alignment points. In other words, using e to represent *b* or *h*, there exists a phrase pair *(e,f)* which satisfies the following constraints:

$$\forall e_i \in e : (e_i, x) \in A \Rightarrow x \in f$$
$$\text{and } \forall f_j \in f : (y, f_j) \in A \Rightarrow y \in e$$
$$\text{and } \exists\, e_i \in e, f_j \in f : (e_i, f_j) \in A$$

where e_i is a word within *e*, f_j is a word within *f* and A is a set of word alignment points. The heuristic allows unaligned words to be included at the boundaries of the source or target phrases.

● *b* and *h* align to the same source phrase *f*, which means there exists an *f* such that both *(b,f)* and *(h,f)* are consistent with the word alignment points.

● *b* and *h* have the same phrase type, such as VP, NP, PP, or other categories.

● *h* must use different words than the backbone phrase b. Note that not all backbone phrases have a paraphrase; the other system translations may use the same phrase.

5 Paraphrase Lattice Construction

The lattice is represented by a target-to-target phrase table consisting of paraphrases and the backbone word order. Given phrase-table format, a standard decoder is able to decode the lattice. Given the backbone and all of its paraphrases, a target-to-target phrase table is constructed using the following steps:

1. Annotate each word in the backbone with its word position information. Using sys1 in Fig 1 as an example, the backbone *B* is modified, resulting in *B'* - "What_1 I_2 ate_3 is_4 noodles_5 other_6 than_7 rice_8". This modified backbone is our decoder input.
2. For each phrase *b* of the backbone, annotate every word in *b* with word position information, resulting in *b'*, such as "other_6 than_7 rice_8".
3. For each phrase *b* of the backbone, we collect every paraphrase *h* of it to add (*b'*, *h*) as an entry to our target-to-target phrase table.
4. For each phrase *b* of the backbone, we add (*b'*, *b*) as an entry to our target-to-target phrase table.

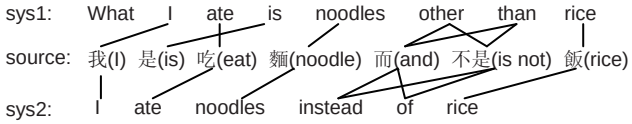

Fig 1. Word alignment of a source sentence and its two hypotheses

Taking the example shown in Fig. 1, the text-to-text phrase table for sys1 translation (backbone) is shown as follows:

(What_1,What)	(I_2,I)	(ate_3,ate)
(I_2 ate_3, I ate)	(What_1 I_2 ate_3, What I ate)	
(is_4, is)	(noodles_5, noodles)	(other_6, other)
(than_7, than)	(rice_8, rice)	
(other_6 than_7 rice_8, other than rice)		
(other_6 than_7 rice_8, instead of rice)		

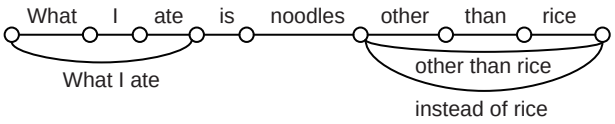

Fig 2 Paraphrase lattice of sys2 translation

Note that in this phrase table, only the last entry is generated by step 3. The remaining entries were generated by step 4.

6 Realization of the Paraphrase Lattice

Any standard decoder could be used to generate all fused hypotheses. The generated hypotheses from sys1 translation (backbone) are

What I ate is noodles other than rice.
What I ate is noodles instead of rice.

The generated hypotheses from the sys2 translation (backbone) are

I ate noodles other than rice.
I ate noodles instead of rice.

Because paraphrases have the same phrase type, the parse of each realization can be obtained by modifying the backbone's parse: replacing the parses of backbone's phrases with the parses of the corresponding paraphrases.

7 Grammatical Error Filtering

Even though syntactic constraints are applied during paraphrase construction, it is still possible that we generate ungrammatical hypotheses, including sentences with agreement or verb mode errors. For example,

sys1: The young student plays basketball
sys2: Many young students play basketball

Assuming "student" and "plays basketball" are paraphrases of "students" and "play basketball", respectively, our generator could provide the following fused hypothesis:

fused hyp: Many young student play basketball

The fused hypothesis then has the agreement problem between "Many", "student" and "play".

7.1 Background

We briefly introduce the FB-LTAG formalism and XTAG English grammar in this section.

7.1.1 Feature-Based Lexicalized Tree Adjoining Grammars

FB-LTAG is based on tree adjoining grammar (TAG) (Joshi et al., 1975). The TAG formalism is a formal tree rewriting system, which consists of a set of elementary trees, corresponding to minimal linguistic structures that localize the dependencies, such as specifying the predicate-argument structure of a lexeme. Elementary trees are divided into initial and auxiliary trees. Initial trees are those for which all non-terminal nodes on the frontier are substitutable, marked with "↓". Auxiliary trees are defined as initial trees, except that exactly one frontier, non-terminal node must be a foot node, marked with "*", with the same label with the root node. Two operations - substitution and adjunction are provided in TAG to adjoin elementary trees.

FB-LTAG has two important characteristics: First, it is a lexicalized TAG. Thus each elementary tree is associated with at least one lexical item. Second, it is a feature-based TAG (Vijay-Shanker and Joshi 1988). Each node in an elementary tree is constrained by two sets of feature-value pairs which are represented by attribute value matrices(AVMs). One AVM (top AVM) defines the relation of the node to its super-tree, and the other AVM (bottom AVM) defines the relation of the node to its descendants. Because of the limited space, we use Fig3 and Fig4 to illustrate the substitution and adjunction operations within the unification framework respectively.

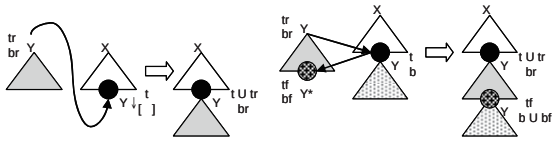

Fig 3 Substitution of FB-LTAG Fig 4 Adjunction of FB-LTAG

7.1.2 XTAG English grammar

XTAG English grammar (XTAG-group 2001) is designed using the FB-LTAG formalism, released[1] by UPENN in 2001. The range of syntactic phenomena that can be handled is large. It defines 57 major elementary trees (tree families) and 50 feature types, such as agreement, case, mode (mood), tense, passive, etc, for its 20,027 lexical entries. Each lexical entry is associated with at least one elementary tree, and each elementary tree is associated with at least one AVM. For example, Fig. 5 shows the simplified elementary trees of "asked". We can see that "asked" specifies its role to be a verb in an indicative sentence (ind), and it should be followed by one indirect object (an NP) and one direct object (a VP) in order. In addition, the direct VP object is restricted to be an infinitive verb phrase (inf).

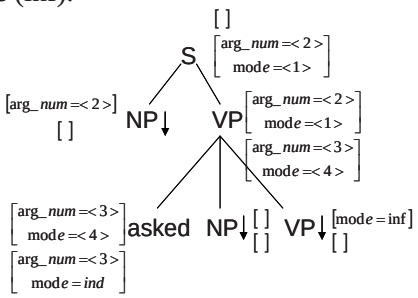

Fig 5 Elementary trees for "asked"

7.2 Grammatical Error Detection

Our procedure for syntactic error detection involves decomposing each sentence hypothesis parse tree into elementary trees, associating each elementary tree with AVMs through look-up in the XTAG grammar, and reconstructing the original parse tree out of the elementary trees using substitution and adjunction operations along with unification of associated AVMs.

When unification of the AVMs fails, a grammatical error has been detected and its error type is identified by the corresponding feature in the AVM.

7.2.1 Decomposing to Elementary trees

Given a translation sentence parse, we decompose it to multiple elementary trees using Chen and Vijay-Shanker's (2000) tree extraction procedure. After that, each lexical item in the sentence will be assigned one elementary tree.

7.2.2 Associating AVMs with Elementary trees

One elementary tree can have multiple possible AVMs associated with it. For example, for the verb "are", one of its elementary trees is associated with three different AVMs, one for 2nd person singular, one for 2nd person plural, and one for 3rd person plural. Unless we can reference the context for "are" (e.g., its subject) we are not sure which

[1] http://www.cis.upenn.edu/~xtag/gramrelease.html

AVM should be used in the reconstruction. So we postpone this decision until later in the reconstruction process. At this point, we associate and record every possible AVM association. Taking the sentence – "Many young student play basketball" as an example, one set of AVM associations for the sentence's extracted elementary trees is as follows:

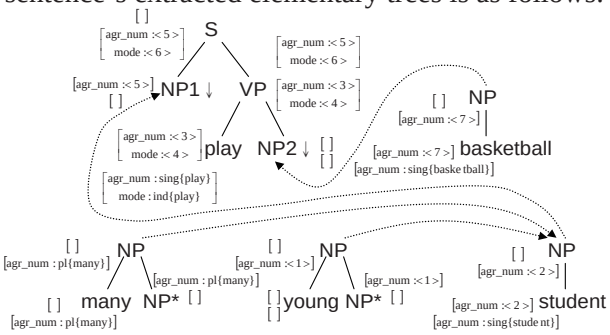

Fig 6 "Many young student play basketball"'s elementary trees with one possible AVMs association. (simplified version)

7.2.3 Reconstruction Framework

Once the elementary trees are associated with AVMs, they will be used to reconstruct the original parse tree through substitution and adjunction operations, illustrated via the dotted lines in Fig 6, to decide if there is any conflict between their AVMs values. It is during this reconstruction process that we detect errors. When a conflict occurs, it will cause an AVM unification failure, reflecting a specific grammatical error.

As we stated in Section 7.2.2, sometimes we are not sure which AVMs for an elementary tree should be used in the reconstruction. So our strategy is to try every possible AVM in order to get the AVM that cause the minimal number of grammatical errors.

In our experiment, 14% generated hypotheses were detected as having grammatical errors and were thus filtered out.

8 Hypothesis Selection

The TER metric is widely used in the backbone selection among system outputs (Rosti et al., 2007; Rosti et al. 2010). In Rosti et al (2007)'s backbone selection, the system translation resulting in the lowest average TER score when aligned against all other system translations is selected as the backbone Es as follows:

$$E_s = \arg\min_i \sum_{j=1}^{N_s} TER(E_j, E_i)$$

$$TER(E_j, E_i) = \frac{Ins + Del + Sub + Shift}{N_i}$$

Where Ei and Ej are two different system translations, Ni is the length of Ei, Ns is the number of systems. In our text-to-text framework, every system translation is regarded as a backbone so it is not necessary to do the backbone selection. However, their idea of using TER as consensus scoring function motivates us to use it to select the best generated hypothesis among our hypothesis pool. In addition, we slightly modify the TER computation as

$$TER(E_j, E_i) = \frac{Ins + Del + Sub + Shift}{N}$$

where N can be any fixed number larger than the maximum length of hypotheses in the pool, making the value of TER to be less than 1. The modification is based on the consideration that while calculating a hypothesis's consensus, the traditional TER would slightly benefit the longer hypotheses and harm the shorter ones. Through this modification, the consensus degree is based only on the number of translation edits, which fairly treat hypotheses with different lengths.

Incorporating system weights, the resulting consensus score function is as follows:

$$ConsScore(E_i) = \sum_{j=1}^{N_s} W_j \times (1 - TER(E_j, E_i))$$

$$E_s = \arg\max_i ConsScore(E_i)$$

where Ns is the number of systems; Ej is a system translation; Ei is a generated hypothesis, and Es is our selection of the best translation. Wj are system weights, which can be tuned through MERT algorithm. In our experiment, we also tried incorporating $ConsScore(E_i)$ with the language model and word penalty of E_i. Their weights are tuned using MERT algorithm.

9 Experiment

Six systems from the DARPA GALE 2008 [2] evaluation were used in the experiments to demon-

[2] The six systems are A: NRC(phrase-based), B: RWTH-PBT(phrase-based), C: RWTH-PBT-AML(phrase-based+source reordering), D: RWTH-PBT-JX(phrase-based+Chinese word segmentation), E: RWTH-PBT-SH(phrase-based+source reordering+rescoring) and F: SRI-HPBT(hiero)

strate the performance of our combination on Chinese-to-English MT tasks. We used a *tune* dataset and *test* dataset for our experiment, each of which is composed of 422 newswire sentences. Every source sentence is provided along with four target references.

In this work, we use Stanford's syntactic parser[3] to parse system translations. Our English language model was trained from 2 billion words. (1-gram# =374165, 2-gram#=9331078, 3-gram#=48944615) The experimental results are shown in Table 3. TTG_filter and TTG_nofilter individually represent the text-to-text generation model with and without grammatical filtering as described in Section 7. TER represents using TER-based consensus score function described in Section 8 to select the best generated hypothesis. TTG_filter+TER+LM represents text-to-text generation with grammatical filtering used with TER and the language model with word penalty to select the best generated hypothesis from the pool.

	tune dataset		*test dataset*	
	BLEU	TER	BLEU	TER
System A	33.97	59.97	32.99	59.31
System B	31.77	64.63	27.95	58.13
System C	34.74	**58.99**	34.40	**58.09**
System D	33.49	62.17	32.96	62.04
System E	**35.33**	59.91	**34.64**	58.67
System F	34.15	59.20	34.13	58.97
TTG_nofilter -TER	**36.73**	**57.33**	**36.07**	**56.63**
TTG_filter -TER	**36.70**	**57.34**	**36.02**	**56.70**
TTG_filter -TER +LM	**36.50**	**57.84**	**35.71**	**57.20**

Table 3 Experimental results

Observing results for *test* in table 3, for TTG_filter -TER, the BLEU score is higher than the best single system by 1.38 points while its WER score is lower than the best single system by 1.39 points, which is a comparable performance with many state-of-the-art CN decoding-based variations in the combination.

We found the performances of TTG_filter-TER and TTG_nofilter-TER are almost the same for both BLEU and TER, but we wondered whether BLEU or TER can really reflect the difference in their translation quality; ungrammatical problems are usually caused by very few words but sometimes they can result in misunderstanding of the entire sentence, especially when those mistakes are caused by verbs. Therefore we carried out a human evaluation task on Amazon Mechanical Turk (AMT) to compare the translation sentences produced by TTG_filter-TER and TTG_nofilter-TER.

32 sentences out of 422 sentences of the *tune* dataset and 37 sentences out of 422 sentences of the *test* dataset produced by TTG_filter-TER and TTG_nofilter-TER are different. So we asked native English speakers on AMT to compare only those translation pairs. The judgment is based on two dimensions separately: *fluency* and *adequacy*. The *fluency* evaluation asked Turk users to judge which translation between the two is more fluent, regardless of the correct meaning of the source, while the *adequacy* evaluation measures which translation between the two conveys the more correct meaning in the source sentence, even if the translation is not fully fluent. For *adequacy*, each comparison (hit) consists of one correct translation reference and the translation pair. For *fluency*, only the translation pair is provided. Each comparison for either *adequacy or fluency* task is done by 5 different native English speakers and the translation with more votes wins.

	tune dataset		*test dataset*	
	better flu	Better ade	better flu	better ade
TTG_nofilter-TER	**47%**	**41%**	**38%**	**41%**
TTG_filter-TER	**53%**	**59%**	**62%**	**59%**

Table 4 Experimental results of human evaluation

The results in Table 4 show that the performance of TTG_filter-TER is better than TTG_nofilter–TER from both the adequacy and the fluency perspectives.

It is interesting that integration with the language model (TTG_filter-TER+LM) performs worse than TTG_filter-TER. We suspect that our text-to-text generation framework has already considered syntactic constraints and thus, taken account of grammatical correctness. The fact that the weight of our language model tuned using MERT algorithm is negative could support this suspicion.

In our experiment (six translation systems), 85% of the realizations for a given backbone's paraphrase lattice will generate less than 300 fused hypotheses. So we set 300 as a threshold for the generation of hypotheses and generate up to 300 for each original system backbone. We found, for our data, that the average number of generated hypotheses for one backbone is around 69. After filtering ungrammatical hypotheses, as described in Section 7, there are, on average, 59 generated hy-

[3] http://nlp.stanford.edu/software/lex-parser.shtml

potheses remaining for one backbone, which means for a given source sentence, around 352 hypotheses are generated. How to improve efficiency is our next step for future work.

10 Analyses of Generated Hypotheses

In this section, we highlight some commonalities and differences between a CN approach and our generation work by taking a closer look at the following two examples. Assume A, B, A*, B*, C are five different phrases and A* and B* are the paraphrases of A and B respectively.

Example1.
sys1: A B sys2: A* C B*
All eight generated hypotheses would be as follows with the top line generated using the backbone- sys1 and the bottom generated using another backbone- sys2.
A B, A B*, B* A, A* B*,
A* C B*, A* C B, A C B*, A C B

Example2.
sys1: A B sys2: B* C A*
All eight generated hypotheses would be
A B, A B*, B* A, A* B*,
B* C A*, B* C A, B C A*, B C A

Under the assumption that every word within a phrase is aligned one-to-one with a word within the corresponding paraphrase, for example 1, the CN will also contain the same eight hypotheses. Observing sys1- "A B" and the fused hypothesis- "A C B", we see that C has been inserted between A and B. This is reasonable, because from sys2- "A* C B*", we know C can be inserted between A* and B* and from this, can infer that C can be inserted into A and B in that order.

But, for example 2, in addition to the eight generated hypotheses, the CN will contain one of the hypotheses- "C A B", "A C B" and "A B C". This is unreasonable because C is only inserted between B* and A* and thus we can only infer that C can be inserted into B and A in that order. A CN allows this possibility and relies on the language model to avoid generating these types of sentences. In contrast, our paraphrase lattice directly avoids the possibility during its construction. Consider an English example where C = "of". We can say "the President of the United States" and "The United States President", but we cannot say "The United States of President".

CN decoding can be regarded as a word-based text-to-text generation framework while ours is a linguistic, phrased-based text-to-text generation framework. CN's large search space enables it to have a higher chance to include the best translation possibility; but, on the other hand, the large search space also raises the risk of generating poor-syntactic hypotheses. The syntactic quality of its realizations mainly relies on the large N-gram language model. Our proposed paraphrase lattice with syntactic constraints has a smaller search space, which could lower the chance to select the really best translation possibility, but the syntactic quality of all realizations within the search space is relatively higher.

11 Related Work

Recently there are also a few phrase-based combination approaches, which can be roughly classified into two types: 1. The phrase lattice is built in the form of a phrase table by aligning phrases of all hypotheses with the source phrases (Rosti et al., 2007; Huang and Papineni 2007), for further re-decoding the source sentence or 2. The phrase lattice is built by aligning phrases of all hypotheses with the phrases of the backbone sentence hypothesis, such as in (Feng et al., 2009) and our work in this paper. Feng et al., (2009) adopt a strategy of extending the traditional confusion network to be a phrase-based lattice and then decoding the lattice to obtain the combination result. The lattice is constructed by incrementally adding alignment hypotheses pairs, which are obtained through a phrase-pair alignment procedure considering several conditions of inserting null words. The main differences between our work and Feng et al., (2009)'s work are that we define our phrases to be linguistic phrases, we apply syntactic constraints in the phrase-pair extraction procedure in order to preserve the syntactic structure of the backbone and we do not insert null words during the lattice construction; critically, we take account of grammatical quality by filtering out ungrammatical generated hypotheses. And finally, our selection of the best hypothesis is postponed to the time after realization in order to use a TER-based metric as the consensus score function, in contrast to their selection during the lattice decoding.

12 Conclusion

Our text-to-text generation approach for system combination features the construction of a paraphrase lattice representing paraphrases of linguistic phrases from a backbone hypothesis sentence. Through the decoding of the paraphrase lattice, we produce all possible hypotheses. A new FB-LTAG-based English grammatical error detector further filters out ungrammatical hypotheses, and a TER-based consensus scoring against all system translations is then used to select the best hypothesis from the pool. Our performance shows the text-to-text generation strategy is a promising and competitive direction for translation combination.

13 Future Work

Improving the efficiency of our approach is one topic for future work. Leusch and Ney (2010) generate 200-best list for advanced re-ranking in their CN framework. It motivates us to investigate the possibility of utilizing an N-best list approach: by attaching each phrase with a simpler consensus score, translation probability or paraphrase probability (Bannard and Callison-Burch 2005) in the paraphrase lattice, one can generate N best hypotheses for later TER-based selection.

Relative to the efficiency issue, on the other hand, our text-to-text generation approach provides the parse of every grammatical generated hypothesis. This enables further advanced rescoring processes, such as checking if the word dependency relations in the source sentence are preserved in each generated hypothesis. This is another direction for future work.

References

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In Proceedings of ACL.

Regina Barzilay and Kathleen McKeown. 2005. Sentence fusion for multidocument summarization. Computational Linguistics, 31.

John Chen and K. Vijay-Shanker. 2000. Automated extraction of TAGs from the Penn treebank. In Proceedings of the Sixth International Workshop on Parsing Technologies.

Jinhua Du, Pavel Pecina and Andy Way. 2010. An Augmented Three-Pass System Combination Framework: DCU Combination System for WMT 2010. In proceedings of the Fifth Workshop on Statistical Machine Translation

Yang Feng, Yang Liu, Haitao Mi, Qun Liu, and Yajuan Lu. 2009 Lattice-based System Combination for Statistical Machine Translation. In Proc. of ACL

Katja Filippova and Michael Strube. Sentence Fusion via Dependency Graph Compression. in the Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing

Xiaodong He, Mei Yang, Jianfeng Gao, Patrick Nguyen, and Robert Moore. 2008. Indirect-hmm-based hypothesis alignment for computing outputs from machine translation systems. In Proc. of EMNLP

Fei Huang and Kishore Papineni. 2007. Hierarchical System Combination for Machine Translation. EMNLP-CoNLL

Aravind K. Joshi, Leon S. Levy, and Masako Takahashi. 1975. Tree Adjunct Grammars. Journal of Computer and System Science, 10:136–163.

Gregor Leusch and Hermann Ney The RWTH System Combination System for WMT 2010. 2010. In proceedings of the Fifth Workshop on Statistical Machine Translation

Erwin Marsi, Emiel Krahmer. 2005. Explorations in Sentence Fusion. In Proceedings of the 10th European Workshop on Natural Language Generation

Evgeny Matusov, Nicola Ueffing, and Hermann Ney 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment, *in Proc. EACL*.

Sushant Narsale JHU System Combination Scheme for WMT 2010. 2010. In proceedings of the Fifth Workshop on Statistical Machine Translation

Franz J. Och and Hermann Ney. 2004. The Alignment Template Approach to Statistical Machine Translation, Computational Linguistics 30(4).

Antti-Veikko I. Rosti, Necip F. Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie J. Dorr. 2007. Combining outputs from multiple machine translation systems. In Proc. of NAACL-HLT

Antti-Veikko Rosti, Bing Zhang, Spyros Matsoukas and Richard Schwartz. 2010. BBN System Description for WMT10 System Combination Task. In proceedings of the Fifth Workshop on Statistical Machine Translation

K. Vijay-Shanker and Aravind K. Joshi. 1988. Feature structure based tree adjoining grammar, in Proceedings of COLING-88, pp. 714-719.

The XTAG-Group. 2001. A Lexicalized Tree Adjoining Grammar for English. Technical Report IRCS 01-03, University of Pennsylvania.