Hierarchical System Combination for Machine Translation

Fei Huang
IBM T.J. Watson Research Center
Yorktown Heights, NY 10562
huangfe@us.ibm.com

Kishore Papineni *
Yahoo! Research
New York, NY 10011
kpapi@yahoo-inc.com

Abstract

Given multiple translations of the same source sentence, how to combine them to produce a translation that is better than any single system output? We propose a hierarchical system combination framework for machine translation. This framework integrates multiple MT systems' output at the word-, phrase- and sentence- levels. By boosting common word and phrase translation pairs, pruning unused phrases, and exploring decoding paths adopted by other MT systems, this framework achieves better translation quality with much less re-decoding time. The full sentence translation hypotheses from multiple systems are additionally selected based on N-gram language models trained on word/word-POS mixed stream, which further improves the translation quality. We consistently observed significant improvements on several test sets in multiple languages covering different genres.

1 Introduction

Many machine translation (MT) frameworks have been developed, including rule-based transfer MT, corpus-based MT (statistical MT and example-based MT), syntax-based MT and the hybrid, statistical MT augmented with syntactic structures. Different MT paradigms have their strengths and weaknesses. Systems adopting the same framework usually produce different translations for the same input, due to their differences in training data, preprocessing, alignment and decoding strategies. It is beneficial to design a framework that combines the decoding strategies of multiple systems as well as their outputs and produces translations better than any single system output. More recently, within the GALE[1] project, multiple MT systems have been developed in each consortium, thus system combination becomes more important.

Traditionally, system combination has been conducted in two ways: glass-box combination and black-box combination. In the glass-box combination, each MT system provides detailed decoding information, such as word and phrase translation pairs and decoding lattices. For example, in the multi-engine machine translation system (Nirenburg and Frederking, 1994), target language phrases from each system and their corresponding source phrases are recorded in a chart structure, together with their confidence scores. A chart-walk algorithm is used to select the best translation from the chart. To combine words and phrases from multiple systems, it is preferable that all the systems adopt similar preprocessing strategies.

In the black-box combination, individual MT systems only output their top-N translation hypotheses without decoding details. This is particularly appealing when combining the translation outputs from COTS MT systems. The final translation may be selected by voted language models and appropriate confidence rescaling schemes ((Tidhar and Kuss-

*This work was done when the author was at IBM Research.

[1] http://www.darpa.mil/ipto/programs/gale/index.htm

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 277–286, Prague, June 2007.

ner, 2000) and (Nomoto, 2004)). (Mellebeek et al., 2006) decomposes source sentences into meaningful constituents, translates them with component MT systems, then selects the best segment translation and combine them based on majority voting, language models and confidence scores.

(Jayaraman and Lavie, 2005) proposed another black-box system combination strategy. Given single top-one translation outputs from multiple MT systems, their approach reconstructs a phrase lattice by aligning words from different MT hypotheses. The alignment is based on the surface form of individual words, their stems (after morphology analysis) and part-of-speech (POS) tags. Aligned words are connected via edges. The algorithm finds the best alignment that minimizes the number of crossing edges. Finally the system generates a new translation by searching the lattice based on alignment information, each system's confidence scores and a language model score. (Matusov et al., 2006) and (Rosti et al., 2007) constructed a confusion network from multiple MT hypotheses, and a consensus translation is selected by redecoding the lattice with arc costs and confidence scores.

In this paper, we introduce our hierarchical system combination strategy. This approach allows combination on word, phrase and sentence levels. Similar to glass-box combination, each MT system provides detailed information about the translation process, such as which source word(s) generates which target word(s) in what order. Such information can be combined with existing word and phrase translation tables, and the augmented phrase table will be significantly pruned according to reliable MT hypotheses. We select an MT system to retranslate the test sentences with the refined models, and encourage search along decoding paths adopted by other MT systems. Thanks to the refined translation models, this approach produces better translations with a much shorter re-decoding time. As in the black-box combination, we select full sentence translation hypotheses from multiple system outputs based on n-gram language models. This hierarchical system combination strategy avoids problems like translation output alignment and confidence score normalization. It seamlessly integrates detailed decoding information and translation hypotheses from multiple MT engines, and produces better translations in an efficient manner. Empirical studies in a later section show that this algorithm improves MT quality by 2.4 BLEU point over the best baseline decoder, with a 1.4 TER reduction. We also observed consistent improvements on several evaluation test sets in multiple languages covering different genres by combining several state-of-the-art MT systems.

The rest of the paper is organized as follows: In section 2, we briefly introduce several baseline MT systems whose outputs are used in the system combination. In section 3, we present the proposed hierarchical system combination framework. We will describe word and phrase combination and pruning, decoding path imitation and sentence translation selection. We show our experimental results in section 4 and conclusions in section 5.

2 Baseline MT System Overview

In our experiments, we take the translation outputs from multiple MT systems. These include phrase-based statistical MT systems (Al-Onaizan and Papineni, 2006) (Block) and (Hewavitharana et al., 2005) (CMU_SMT) , a direct translation model (DTM) system (Ittycheriah and Roukos, 2007) and a hierarchical phrased-based MT system (Hiero) (Chiang, 2005). Different translation frameworks are adopted by different decoders: the DTM decoder combines different features (source words, morphemes and POS tags, target words and POS tags) in a maximum entropy framework. These features are integrated with a phrase translation table for flexible distortion model and word selection. The CMU_SMT decoder extracts testset-specific bilingual phrases on the fly with PESA algorithm. The Hiero system extracts context-free grammar rules for long range constituent reordering.

We select the IBM block decoder to re-translate the test set for glass-box system combination. This system is a multi-stack, multi-beam search decoder. Given a source sentence, the decoder tries to find the translation hypothesis with the minimum translation cost. The overall cost is the log-linear combination of different feature functions, such as translation model cost, language model cost, distortion cost and sentence length cost. The translation cost between a phrase translation pair (f, e) is defined as

$$TM(e, f) = \sum_i \lambda_i \phi(i) \quad (1)$$

where feature cost functions $\phi(i)$ includes:

$-\log p(f|e)$, a target-to-source word translation cost, calculated based on unnormalized IBM model1 cost (Brown et al., 1994);

$$p(f|e) = \prod_j \sum_i t(f_j|e_i) \quad (2)$$

where $t(f_j|e_i)$ is the word translation probabilities, estimated based on word alignment frequencies over all the training data. i and j are word positions in target and source phrases.

$-\log p(e|f)$, a source-to-target word translation cost, calculated similar to $-\log p(f|e)$;

$S(e, f)$, a phrase translation cost estimated according to their relative alignment frequency in the bilingual training data,

$$S(e, f) = -\log P(e|f) = -\log \frac{C(f, e)}{C(f)}. \quad (3)$$

λ's in Equation 1 are the weights of different feature functions, learned to maximize development set BLEU scores using a method similar to (Och, 2003).

The SMT system is trained with testset-specific training data. This is not cheating. Given a test set, from a large bilingual corpora we select parallel sentence pairs covering n-grams from source sentences. Phrase translation pairs are extracted from the subsampled alignments. This not only reduces the size of the phrase table, but also improves topic relevancy of the extracted phrase pairs. As a results, it improves both the efficiency and the performance of machine translation.

3 Hierarchical System Combination Framework

The overall system combination framework is shown in Figure 1. The source text is translated by multiple baseline MT systems. Each system produces both top-one translation hypothesis as well as phrase pairs and decoding path during translation. The information is shared through a common XML file format, as shown in Figure 2. It demonstrates how a source sentence is segmented into a sequence of phrases, the order and translation of each source phrase as well as the translation scores, and a vector of feature scores for the whole test sentence. Such XML files are generated by all the systems when they translate the source test set.

We collect phrase translation pairs from each decoder's output. Within each phrase pair, we identify word alignment and estimate word translation probabilities. We combine the testset-specific word translation model with a general model. We augment the baseline phrase table with phrase translation pairs extracted from system outputs, then prune the table with translation hypotheses. We retranslate the source text using the block decoder with updated word and phrase translation models. Additionally, to take advantage of flexible reordering strategies of other decoders, we develop a word order cost function to reinforce search along decoding paths adopted by other decoders. With the refined translation models and focused search space, the block decoder efficiently produces a better translation output. Finally, the sentence hypothesis selection module selects the best translation from each systems' top-one outputs based on language model scores. Note that the hypothesis selection module does not require detailed decoding information, thus can take in any MT systems' outputs.

3.1 Word Translation Combination

The baseline word translation model is too general for the given test set. Our goal is to construct a testset-specific word translation model, combine it with the general model to boost consensus word translations. Bilingual phrase translation pairs are read from each system-generated XML file. Word alignments are identified within a phrase pair based on IBM Model-1 probabilities. As the phrase pairs are typically short, word alignments are quite accurate. We collect word alignment counts from the whole test set translation, and estimate both source-to-target and target-to-source word translation probabilities. We combine such testset-specific translation model with the general model.

$$t''(e|f) = \gamma t'(e|f) + (1 - \gamma) t(e|f); \quad (4)$$

where $t'(e|f)$ is the testset-specific source-to-target word translation probability, and $t(e|f)$ is the prob-

```
<tr engine="XXX">
<s id="0"> <w> اردوغان </w><w> يؤكد </w><w> بان </w><w> تركيا </w><w> سترفض
</w><w> اي </w><w> ضغوطات </w><w> لحثها </w><w> علي </w><w> الاعتراف </w><w>
بقبرص </w></s>
<hyp r="0" c="2.15357">
 <t>
<p al="0-0" cost="0.0603734"> erdogan </p>
<p al="1-1" cost="0.367276"> emphasized </p>
<p al="2-2" cost="0.128066"> that </p>
<p al="3-3" cost="0.0179338"> turkey </p>
<p al="4-5" cost="0.379862"> would reject any </p>
<p al="6-6" cost="0.221536"> pressure </p>
<p al="7-7" cost="0.228264"> to urge them </p>
<p al="8-8" cost="0.132242"> to</p>
<p al="9-9" cost="0.113983"> recognize </p>
<p al="10-10" cost="0.133359"> Cyprus </p>
</t>
<sco>
19.6796 8.40107 0.333514 0.00568583 0.223554 0 0.352681 0.01 -0.616 0.009 0.182052
</sco>
</hyp>
</tr>
```


Figure 2: Sample XML file format. This includes a source sentence (segmented as a sequence of source phrases), their translations as well as a vector of feature scores (language model scores, translation model scores, distortion model scores and a sentence length score).

ability from general model. γ is the linear combination weight, and is set according to the confidence on the quality of system outputs. In our experiments, we set γ to be 0.8. We combine both source-to-target and target-to-source word translation models, and update the word translation costs, $-\log p(e|f)$ and $-\log p(f|e)$, accordingly.

3.2 Phrase Translation Combination and Pruning

Phrase translation pairs can be combined in two different ways. We may collect and merge testset-specific phrase translation tables from each system, if they are available. Essentially, this is similar to combining the training data of multiple MT systems. The new phrase translation probability is calculated according to the updated phrase alignment frequencies:

$$P'(e|f) = \frac{C_b(f,e) + \sum \alpha_m C_m(f,e)}{C_b(f) + \sum \alpha_m C_m(f)}, \quad (5)$$

where C_b is the phrase pair count from the baseline block decoder, and C_m is the count from other MT systems. α_m is a system-specific linear combination weight. If not all the phrase tables are available, we collect phrase translation pairs from system outputs, and merge them with C_b. In such case, we may adjust α to balance the small counts from system outputs and large counts from C_b.

The corresponding phrase translation cost is updated as

$$S'(e,f) = -\log P'(e|f). \quad (6)$$

Another phrase combination strategy works on the sentence level. This strategy relies on the consensus of different MT systems when translating the same source sentence. It collects phrase translation pairs used by different MT systems to translate the same sentence. Similarly, it boosts common phrase pairs that are selected by multiple decoders.

$$S''(e,f) = \frac{\beta}{|C(f,e)|} \times S'(e,f), \quad (7)$$

where β is a boosting factor, $0 < \beta \leq 1$. $|C(f,e)|$ is the number of systems that use phrase pair (f,e) to translate the input sentence. A phrase translation pair selected by multiple systems is more likely a good translation, thus costs less.

The combined phrase table contains multiple translations for each source phrase. Many of them are unlikely translations given the context. These phrase pairs produce low-quality partial hypotheses during hypothesis expansion, incur unnecessary model cost calculation and larger search space, and reduce the translation efficiency. More importantly, the translation probabilities of correct phrase pairs are reduced as some probability mass is distributed among incorrect phrase pairs. As a result, good phrase pairs may not be selected in the final translation.

Oracle experiments show that if we prune the phrase table and only keep phrases that appear in the reference translations, we can improve the translation quality by 10 BLEU points. This shows the potential gain by appropriate phrase pruning. We developed a phrase pruning technique based on self-training. This approach reinforces phrase translations learned from MT system output. Assuming we have reasonable first-pass translation outputs, we only keep phrase pairs whose target phrase is covered by existing system translations. These phrase pairs include those selected in the final translations, as well as their combinations or sub-phrases. As a result, the size of the phrase table is reduced by 80-90%, and the re-decoding time is reduced by 80%. Because correct phrase translations are assigned higher probabilities, it generates better translations with higher BLEU scores.

3.3 Decoding Path Imitation

Because of different reordering models, words in the source sentence can be translated in different orders. The block decoder has local reordering capability that allows source words within a given window to jump forward or backward with a certain cost. The DTM decoder takes similar reordering strategy, with some variants like dynamic window width depending on the POS tag of the current source word. The Hiero system allows for long range constituent reordering based on context-free grammar rules. To combine different reordering strategies from various decoders, we developed a reordering cost function that encourages search along decoding paths adopted by other decoders.

From each system's XML file, we identify the order of translating source words based on word alignment information. For example, given the following hypothesis path,

<p al="0-1"> izzat ibrahim </p> <p al="2-2"> receives </p> <p al="3-4"> an economic official </p> <p al="5-6"> in </p> <p al="7-7"> baghdad </p>

We find the source phrase containing words [0,1] is first translated into a target phrase "*izzat ibrahim*", which is followed by the translation from source word 2 to a single target word "*receives*", etc.. We identify the word alignment within the phrase translation pairs based on IBM model-1 scores. As a result, we get the following source word translation sequence from the above hypothesis (note: source word 5 is translated as NULL):

$0 < 1 < 2 < 4 < 3 < 6 < 7$

Such decoding sequence determines the translation order between any source word pairs, e.g., word 4 should be translated before word 3, 6 and 7. We collect such ordered word pairs from all system outputs' paths. When re-translating the source sentence, for each partially expanded decoding path, we compute the ratio of word pairs that satisfy such ordering constraints[2].

Specifically, given a partially expanded path $P = \{s_1 < s_2 < \cdots < s_m\}$, word pair $(s_i < s_j)$ implies s_i is translated before s_j. If word pair $(s_i < s_j)$ is covered by a full decoding path Q (from other system outputs), we denote the relationship as $(s_i < s_j) \in Q$.

For any ordered word pair $(s_i < s_j) \in P$, we define its matching ratio as the percentage of full decoding paths that cover it:

$$R(s_i < s_j) = \frac{|Q|}{N}, \{Q|(s_i < s_j) \in Q\} \quad (8)$$

where N is the total number of full decoding paths.

We define the path matching cost function:

$$L(P) = -\log \frac{\sum_{\forall (s_i < s_j) \in P} R(s_i < s_j)}{\sum_{\forall (s_i < s_j) \in P} 1} \quad (9)$$

The denominator is the total number of ordered word pairs in path P. As a result, partial paths are boosted if they take similar source word translation orders as other system outputs. This cost function is multiplied with a manually tuned model weight before integrating into the log-linear cost model framework.

[2]We set no constraints for source words that are translated into NULL.

3.4 Sentence Hypothesis Selection

The sentence hypothesis selection module only takes the final translation outputs from individual systems, including the output from the glass-box combination. For each input source sentence, it selects the "optimal" system output based on certain feature functions.

We experiment with two feature functions. One is a typical 5-gram word language model (LM). The optimal translation output E' is selected among the top-one hypothesis from all the systems according to their LM scores. Let e_i be a word in sentence E:

$$\begin{aligned} E' &= \arg\min_E -\log P_{5glm}(E) \quad (10) \\ &= \arg\min_E \sum_i -\log p(e_i|e_{i-4}^{i-1}), \end{aligned}$$

where e_{i-4}^{i-1} is the n-gram history, $(e_{i-4}, e_{i-3}, e_{i-2}, e_{i-1})$.

Another feature function is based on the 5-gram LM score calculated on the mixed stream of word and POS tags of the translation output. We run POS tagging on the translation hypotheses. We keep the word identities of top N frequent words (N=1000 in our experiments), and the remaining words are replaced with their POS tags. As a result, the mixed stream is like a skeleton of the original sentence, as shown in Figure 3.

With this model, the optimal translation output E^* is selected based on the following formula:

$$\begin{aligned} E^* &= \arg\min_E -\log P_{wplm}(E) \quad (11) \\ &= \arg\min_E \sum_i -\log p(T(e_i)|T(e)_{i-4}^{i-1}) \end{aligned}$$

where the mixed stream token $T(e) = e$ when $e \leq N$, and $T(e) = POS(e)$ when $e > N$. Similar to a class-based LM, this model is less prone to data sparseness problems.

4 Experiments

We experiment with different system combination strategies on the NIST 2003 Arabic-English MT evaluation test set. Testset-specific bilingual data are subsampled, which include 260K sentence pairs, 10.8M Arabic words and 13.5M English words. We report case-sensitive BLEU (Papineni et al., 2001) and TER (Snover et al., 2006) as the MT evaluation metrics. We evaluate the translation quality of different combination strategies:

- **WdCom:** Combine testset-specific word translation model with the baseline model, as described in section 3.1.
- **PhrCom:** Combine and prune phrase translation tables from all systems, as described in section 3.2. This include testset-specific phrase table combination (**Tstcom**), sentence level phrase combination (**Sentcom**) and phrase pruning based on translation hypotheses (**Prune**).
- **Path:** Encourage search along the decoding paths adopted by other systems via path matching cost function, as described in section 3.3.
- **SenSel:** Select whole sentence translation hypothesis among all systems' top-one outputs based on N-gram language models trained on word stream (**word**) and word-POS mixed stream(**wdpos).**

	BLEUr4n4c	TER
sys1	0.5323	43.11
sys4	0.4742	46.35
Tstcom	0.5429	42.64
Tstcom+Sentcom	0.5466	42.32
Tstcom+Sentcom+Prune	0.5505	42.21

Table 1: Translation results with phrase combination and pruning.

Table 1 shows the improvement by combining phrase tables from multiple MT systems using different combination strategies. We only show the highest and lowest baseline system scores. By combining testset-specific phrase translation tables (**Tstcom**), we achieved 1.0 BLEU improvement and 0.5 TER reduction. Sentence-level phrase combination and pruning additionally improve the BLEU score by 0.7 point and reduce TER by 0.4 percent.

Table 2 shows the improvement with different sentence translation hypothesis selection approaches. The word-based LM is trained with about 1.75G words from newswire text. A distributed

	BLEUr4n4c	TER
sys1	0.5323	43.11
sys2	0.5320	43.06
SentSel-word:	0.5354	42.56
SentSel-wpmix:	0.5380	43.06

Table 2: Translation results with different sentence hypothesis selection strategies.

	BLEUr4n4c	TER
sys1	0.5323	43.11
sys2	0.5320	43.06
sys3	0.4922	46.03
sys4	0.4742	46.35
WdCom	0.5339	42.60
WdCom+PhrCom	0.5528	41.98
WdCom+PhrCom+Path	0.5543	41.75
WdCom+PhrCom+Path+SenSel	0.5565	41.59

Table 3: Translation results with hierarchical system combination strategy.

	BLEUr4n4c	TER
sys1	0.3205	60.48
sys2	0.3057	59.99
sys3	0.2787	64.46
sys4	0.2823	59.19
sys5	0.3028	62.16
syscom	0.3409	58.89

Table 4: System combination results on Chinese-English translation.

	BLEUr1n4c	TER
sys1	0.1261	71.70
sys2	0.1307	77.52
sys3	0.1282	70.82
sys4	0.1259	70.20
syscom	0.1386	69.23

Table 5: System combination results for Arabic-English web log translation.

large-scale language model architecture is developed to handle such large training corpora[3], as described in (Emami et al., 2007). The word-based LM shows both improvement in BLEU scores and error reduction in TER. On the other hand, even though the word-POS LM is trained with much less data (about 136M words), it improves BLEU score more effectively, though there is no change in TER.

Table 3 shows the improvements from hierarchical system combination strategy. We find that word-based translation combination improves the baseline block decoder by 0.16 BLEU point and reduce TER by 0.5 point. Phrase-based translation combination (including phrase table combination, sentence-level phrase combination and phrase pruning) further improves the BLEU score by 1.9 point (another 0.6 drop in TER). By encouraging the search along other decoder's decoding paths, we observed additional 0.15 BLEU improvement and 0.2 TER reduction. Finally, sentence translation hypothesis selection with word-based LM led to 0.2 BLEU point improvement and 0.16 point reduction in TER. To summarize, with the hierarchical system combination framework, we achieved 2.4 BLEU point improvement over the best baseline system, and reduce the TER by 1.4 point.

Table 4 shows the system combination results on Chinese-English newswire translation. The test data is NIST MT03 Chinese-English evaluation test set. In addition to the 4 baseline MT systems, we also add another phrase-based MT system (Lee et al., 2006). The system combination improves over the best baseline system by 2 BLEU points, and reduce the TER score by 1.6 percent. Thanks to the long range constituent reordering capability of different baseline systems, the path imitation improves the BLEU score by 0.4 point.

We consistently notice improved translation quality with system combination on unstructured text and speech translations, as shown in Table 5 and 6. With one reference translation, we notice 1.2 BLEU point improvement over the baseline block decoder (with 2.5 point TER reduction) on web log translation and about 2.1 point BLEU improvement (with 0.9 point TER reduction) on Broadcast News speech translation.

[3]The same LM is also used during first pass decoding by both the block and the DTM decoders.

	BLEUr1n4c	TER
sys1	0.2011	61.46
sys2	0.2211	66.32
sys3	0.2074	61.21
sys4	0.1258	85.45
syscom	0.2221	60.54

Table 6: System combination results for Arabic-English speech translation.

5 Related Work

Many system combination research have been done recently. (Matusov et al., 2006) computes consensus translation by voting on a confusion network, which is created by pairwise word alignment of multiple baseline MT hypotheses. This is similar to the sentence- and word- level combinations in (Rosti et al., 2007), where TER is used to align multiple hypotheses. Both approaches adopt black-box combination strategy, as target translations are combined independent of source sentences. (Rosti et al., 2007) extracts phrase translation pairs in the phrase level combination. Our proposed method incorporates bilingual information from source and target sentences in a hierarchical framework: word, phrase and decoding path combinations. Such information proves very helpful in our experiments. We also developed a path matching cost function to encourage decoding path imitation, thus enable one decoder to take advantage of rich reordering models of other MT systems. We only combine top-one hypothesis from each system, and did not apply system confidence measure and minimum error rate training to tune system combination weights. This will be our future work.

6 Conclusion

Our hierarchical system combination strategy effectively integrates word and phrase translation combinations, decoding path imitation and sentence hypothesis selection from multiple MT systems. By boosting common word and phrase translation pairs and pruning unused ones, we obtain better translation quality with less re-decoding time. By imitating the decoding paths, we take advantage of various reordering schemes from different decoders. The sentence hypothesis selection based on N-gram language model further improves the translation quality. The effectiveness has been consistently proved in several empirical studies with test sets in different languages and covering different genres.

7 Acknowledgment

The authors would like to thank Yaser Al-Onaizan, Abraham Ittycheriah and Salim Roukos for helpful discussions and suggestions. This work is supported under the DARPA GALE project, contract No. HR0011-06-2-0001.

References

Yaser Al-Onaizan and Kishore Papineni. 2006. Distortion Models for Statistical Machine Translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 529–536, Sydney, Australia, July. Association for Computational Linguistics.

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1994. The Mathematic of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

David Chiang. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Ahmad Emami, Kishore Papineni, and Jeffrey Sorensen. 2007. Large-scale Distributed Language Modeling. In *Proceedings of the 2007 International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2007)*, Honolulu, Hawaii, April.

Sanjika Hewavitharana, Bing Zhao, Almut Silja Hildebrand, Matthias Eck, Chiori Hori, Stephan Vogel, and Alex Waibel. 2005. The CMU Statistical Machine Translation System for IWSLT2005. In *Proceedings of IWSLT 2005*, Pittsburgh, PA, USA, November.

Arraham Ittycheriah and Salim Roukos. 2007. Direct Translation Model2. In *Proceedings of the 2007 Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2007)*, Rochester, NY, April. Association for Computational Linguistics.

Shyamsundar Jayaraman and Alon Lavie. 2005. Multi-Engine Machine Translation Guided by Explicit Word Matching. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 101–104, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Y-S. Lee, S. Roukos, Y. Al-Onaizan, and K. Papineni. 2006. IBM Spoken Language Translation System. In *Proc. of TC-STAR Workshop on Speech-to-Speech Translation*, Barcelona, Spain.

Evgeny Matusov, Nicola Ueffing, and Hermann Ney. 2006. Computing Consensus Translation for Multiple Machine Translation Systems Using Enhanced Hypothesis Alignment. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL '06)*, pages 263–270, Trento, Italy, April. Association for Computational Linguistics.

B. Mellebeek, K. Owczarzak, J. Van Genabith, and A. Way. 2006. Multi-Engine Machine Translation by Recursive Sentence Decomposition. In *Proceedings of the 7th biennial conference of the Association for Machine Translation in the Americas*, pages 110–118, Boston, MA, June.

Sergei Nirenburg and Robert Frederking. 1994. Toward Multi-engine Machine Translation. In *HLT '94: Proceedings of the workshop on Human Language Technology*, pages 147–151, Morristown, NJ, USA. Association for Computational Linguistics.

Tadashi Nomoto. 2004. Multi-Engine Machine Translation with Voted Language Model. In *Proceedings of ACL*, pages 494–501.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of ACL*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: a Method for Automatic Evaluation of Machine Translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Antti-Veikko Rosti, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie J. Dorr. 2007. Combining Translations from Multiple Machine Translation Systems. In *Proceedings of the Conference on Human Language Technology and North American chapter of the Association for Computational Linguistics Annual Meeting (HLT-NAACL'2007)*, Rochester, NY, April.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of Association for Machine Translation in the Americas*.

D. Tidhar and U. Kussner. 2000. Learning to Select a Good Translation. In *Proceedings of the International Conference on Computational Linguistics*, pages 843–849.

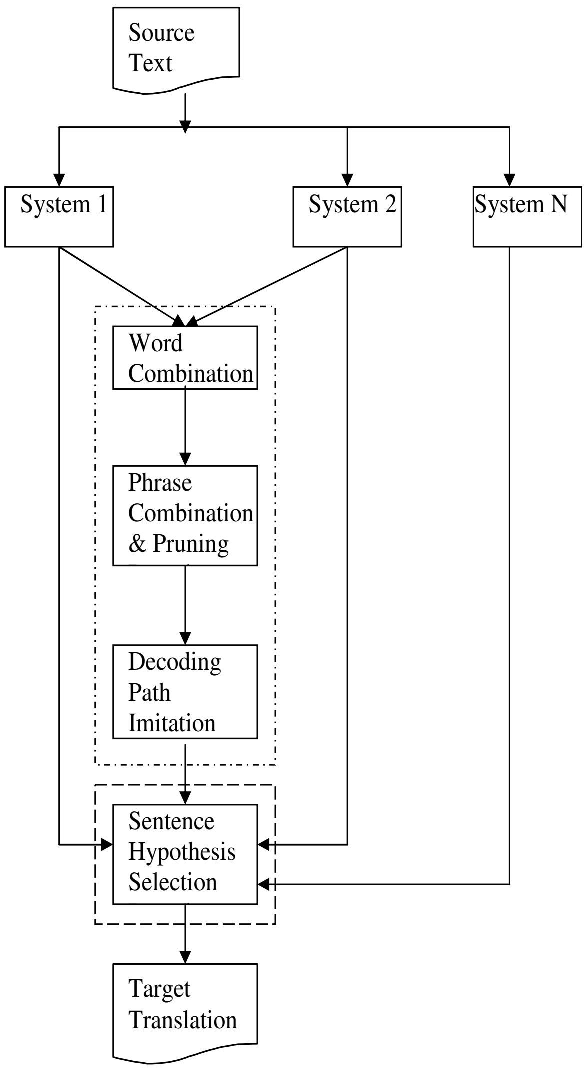

Figure 1: Hierarchical MT system combination architecture. The top dot-line rectangle is similar to the glass-box combination, and the bottom rectangle with sentence selection is similar to the black-box combination.

Original Sentence:

in *short* , making a good plan at the *beginning* of the construction is the *crucial measure* for *reducing haphazard* economic development .

Word-POS mixed stream:

in JJ , making a good plan at the NN of the construction is the JJ NN for VBG JJ economic development .

Figure 3: Sentence with Word-POS mixed stream.