Voting on N-grams for Machine Translation System Combination

Kenneth Heafield and Alon Lavie
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{heafield,alavie}@cs.cmu.edu

Abstract

System combination exploits differences between machine translation systems to form a combined translation from several system outputs. Core to this process are features that reward n-gram matches between a candidate combination and each system output. Systems differ in performance at the n-gram level despite similar overall scores. We therefore advocate a new feature formulation: for each system and each small n, a feature counts n-gram matches between the system and candidate. We show post-evaluation improvement of 6.67 BLEU over the best system on NIST MT09 Arabic-English test data. Compared to a baseline system combination scheme from WMT 2009, we show improvement in the range of 1 BLEU point.

1 Introduction

System combination merges the output of several machine translation systems to form an improved translation. While individual systems perform similarly overall, human evaluators report different error distributions for each system (Peterson et al., 2009b). For example, some systems are weak at word order while others have more trouble with nouns and verbs. Existing system combination techniques (Rosti et al., 2008; Karakos et al., 2008; Leusch et al., 2009a) ignore these distinctions by learning a single weight for each system. This weight is used in word-level decisions and therefore captures only lexical choice. We see two problems: system behavior differs in more ways than captured by a single weight and further current features only guide decisions at the word level. To remedy this situation, we propose new features that account for multiword behavior, each with a separate set of system weights.

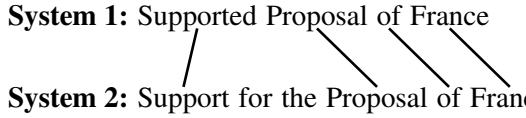
2 Features

Most combination schemes generate many hypothesis combinations, score them using a battery of features, and search for a hypothesis with highest score to output. Formally, the system generates hypothesis h, evaluates feature function f, and multiplies linear weight vector λ by feature vector $f(h)$ to obtain score $\lambda^T f(h)$. The score is used to rank final hypotheses and to prune partial hypotheses during beam search. The beams contain hypotheses of equal length. With the aim of improving score and therefore translation quality, this paper focuses on the structure of features f and their corresponding weights λ.

The feature function f consists of the following feature categories:

Length Length of the hypothesis, as in Koehn et al. (2007). This compensates, to first order, for the impact of length on other features.

LM Log probability from an SRI (Stolcke, 2002) language model. When the language model scores a word, it finds the longest n-gram in the model with the same word and context. We use the length n as a second feature. The purpose of this second feature is to provide the scoring model with limited control over language model backoff penalties.



Candidate: Support for Proposal of France

	Unigram	Bigram	Trigram
System 1	4	2	1
System 2	5	3	1

Figure 1: Example match feature values with two systems and matches up to length $l = 3$. Here, "Supported" counts because it aligns with "Support".

Match The main focus of this paper, this feature category captures how well the hypothesis corresponds to the system outputs. It consists of several features of the form $c_{s,n}$ which count n-grams in the hypothesis that match the sentence output by system s. We use several instantiations of feature $c_{s,n}$, one for each system s and for length n ranging from 1 to some maximum length l. The hyper parameter l is considered in Section 6.1. Figure 1 shows example values with two systems and $l = 3$. The number of match features is l times the number of systems being combined.

The match features consider only the specific sentences being combined, all of which are translations from a single source sentence. By contrast, Leusch et al. (2009a) count matches to the entire translated corpus. While our approach considers the most specific data available, theirs has the advantage of gathering more data with which to compare, including document-level matches. As discussed in Section 3, we also differ significantly from Leusch et al. (2009a) in that our features have tunable system and n-gram weights while theirs are hyper parameters.

Together, these define the feature function f. We focus primarily on the new match features that capture lexical and multiword agreement with each system. The weight on match count $c_{s,n}$ corresponds to confidence in n-grams from system s. However, this weight also accounts for correlation between features, which is quite high within the same system and across related systems. Viewed as language modeling, each $c_{s,n}$ is a miniature language model trained on the translated sentence output by system s and jointly interpolated with a traditional language model and with peer models.

As described further in Section 4, we jointly tune the weights λ using modified minimum error rate training (Och, 2003). In doing so, we simultaneously learn several weights for each system, one for each length n-gram output by that system. The unigram weight captures confidence in lexical choice while weights on longer n-gram features capture confidence in word order and phrasal choices. These features are most effective with a variety of hypotheses from which to choose, so in Section 5 we describe a search space with more flexible word order. Combined, these comprise a combination scheme that differs from others in three key ways: the search space is more flexible, the features consider matches longer than unigrams, and system weights differ by task.

3 Related Work

System combination takes a variety of forms that pair a space of hypothesis combinations with features to score these hypotheses. Here, we are primarily interested in three aspects of each combination scheme: the space of hypotheses, features that reward n-gram matches with system outputs, and the system weights used for those features.

3.1 Search Spaces

Hypothesis selection (Hildebrand and Vogel, 2009) and minimum Bayes risk (Kumar and Byrne, 2004) select from k-best lists output by each system. In the limit case for large k, DeNero et al. (2010) adopt the search spaces of the translation systems being combined.

Confusion networks preserve the word order of one k-best list entry called the backbone. The backbone is chosen by hypothesis selection (Karakos et al., 2008; Sim et al., 2007) or jointly with decoding (Leusch et al., 2009a; Rosti et al., 2008). Other k-best entries are aligned to the backbone. The search space consists of choosing each word from among the alternatives aligned with it, keeping these in backbone order. With this search space, the im-

pact of our match features is limited to selection at the word level, multiword lexical choice, and possibly selection of the backbone.

Flexible ordering schemes use a reordering model (He and Toutanova, 2009) or dynamically switch backbones (Heafield et al., 2009) to create word orders not seen in any single translation. For example, these translations appear in NIST MT09 (Peterson et al., 2009a): *"We have not the interest of control on the Palestinians life,"* and *"We do not have a desire to control the lives of the Palestinians."* Flexible ordering schemes consider *"have not"* versus *"do not have"* separately from *"Palestinians life"* versus *"lives of the Palestinians."* Our match features have the most impact here because word order is less constrained. This is the type of search space we use in our experiments.

3.2 N-gram Match Features

Agreement is central to system combination and most schemes have some form of n-gram match features. Of these, the simplest consider only unigram matches (Ayan et al., 2008; Heafield et al., 2009; Rosti et al., 2008; Zhao and Jiang, 2009).

Some schemes go beyond unigrams but with fixed weight. Karakos (2009) uses n-gram matches to select the backbone but only unigrams for decoding. Kumar and Byrne (2004) use arbitrary evaluation metric to measure similarity. BLEU (Papineni et al., 2002) is commonly used for this purpose and quite similar to our match features, although we have tunable linear n-gram and length weights instead of fixed geometric weights.

Several schemes expose a separate feature for each n-gram length (Hildebrand and Vogel, 2009; Leusch et al., 2009a; Zens and Ney, 2006; Zhao and He, 2009). Some of these are conceptualized as a language model that, up to edge effects, exposes the log ratio of $(n+1)$-gram matches to n-gram matches. An equivalent linear combination of these features exposes the log n-gram match counts directly. These separate features enable tuning n-gram weights.

3.3 System Weighting

Different and correlated system strengths make it important to weight systems when combining their votes on n-grams. The simplest method treats these system weights as a hyper parameter (Heafield et al., 2009; Hildebrand and Vogel, 2009). The hyper parameter might be set to an increasing function of each system's overall performance (Rosti et al., 2009; Zhao and Jiang, 2009); this does not account for correlation.

Some combination schemes (Karakos et al., 2008; Leusch et al., 2009a; Rosti et al., 2008; Zens and Ney, 2006) tune system weights, but only as they apply to votes on unigrams. In particular, Leusch et al. (2009a) note that their system weights impact neither their trigram language model based on system outputs nor selection of a backbone. We answer these issues by introducing separate system weights for each n-gram length in the system output language model and by not restricting search to the order of a single backbone.

Our experiments use only 1-best outputs, so system-level weights suffice here. For methods that use k-best lists, system weight may be moderated by some decreasing function of rank in the k-best list (Ayan et al., 2008; Zhao and He, 2009). Minimum Bayes risk (Kumar and Byrne, 2004) takes the technique a step further by using the overall system scores that determined the ranking.

4 Parameter Tuning

Key to our model is jointly tuning the feature weights λ. In our experiments, weight vector λ is tuned using minimum error rate training (MERT) (Och, 2003) towards BLEU (Papineni et al., 2002). We also tried tuning towards TER minus BLEU (Rosti et al., 2007) and METEOR (Lavie and Denkowski, 2010), finding at best minor improvement in the targeted metric with longer tuning time. This may be due to underlying systems tuning primarily towards BLEU.

In ordinary MERT, the decoder produces hypotheses given weights λ and the optimizer selects λ to rank the best hypotheses at the top. These steps alternate until λ converges or enough iterations happen. As the feature weights converge, the k-best lists output also converge. In our experiments, we use $k = 300$. For long sentences, this is a small fraction of the hypotheses that our flexible ordering scheme can generate using 1-best outputs from each system. The problem here is that the decoder sees mostly the

same λ each time and the optimizer sees mostly the same output each time, missing potentially better but different weights. Random restarts inside the optimizer do not solve this problem because this technique only finds better weights subject to decoded hypotheses. As the number of features increases (in some experiments to 39), the problem becomes more severe because the space of feature weights is much larger than the explored space.

We propose a simulated annealing method to address problems with MERT, leaving other tuning methods such as MIRA (Chiang et al., 2008) and lattice MERT to future work. Specifically, when the decoder is given weights λ to use for decoding in iteration $0 \leq j < 10$, it instead uses weights μ sampled according to

$$\mu_i \sim U\left[\frac{j}{10}\lambda_i, \left(2 - \frac{j}{10}\right)\lambda_i\right]$$

where U is the uniform distribution and subscript i denotes the ith feature. This sampling is done on a per-sentence basis, so the first sentence is decoded with different weights than the second sentence. The amount of random perturbation decreases linearly each iteration until the 10th and subsequent iterations where weights are used in the normal, unperturbed, fashion. The process therefore converges to normal minimum error rate training. In practice, this technique increases the number of iterations and decreases the difference in tuning scores following MERT. The specific formulation may not be optimal, but suffices for our goal of tuning 39 feature weights.

5 Combination Scheme

We use our decoder (Heafield et al., 2009) with some modifications. The process starts by aligning single best translations in pairs using METEOR (Lavie and Denkowski, 2010). In decreasing order of priority, words are aligned exactly, by shared stem (Porter, 2001), by shared WordNet (Fellbaum, 1998) synset, or according to unigram paraphrases from the TERp (Snover et al., 2008) database.

Search proceeds inductively using the aligned translations. The initial hypothesis consists of the beginning of sentence tag. A hypothesis branches into several hypotheses by appending the first unused word from any system. This word, and those aligned with it, are marked as used in the hypothesis. Essentially, the hypothesis strings together non-overlapping fragments from each system. Choosing one fragment e.g. *"Palestinians life"* over another *"lives of the Palestinians"* leaves the unaligned words behind. A heuristic, described fully in Heafield et al. (2009), detects when a system falls too far behind as a result and skips such words to maintain synchronization.

Alignments also define the tolerance of match features; we experiment with alignment types accepted by the match features in Section 6.3. Specifically, a system's unigram match count includes both words taken from or aligned with the system. Bigram matches consist of two consecutive unigram matches with the same word order; higher order n-grams matches are similar.

5.1 Baseline System

Our cmu-combo submission to the 2009 Workshop on Machine Translation (Heafield et al., 2009) has the same search space but simpler features, so it serves as a controlled baseline. Nonetheless, this baseline was judged best, or insignificantly different from best, in official human judgments performed as part of the evaluation (Callison-Burch et al., 2009) for every scenario considered in this paper.

The baseline scheme has a single unigram match feature using hyper parameter system weights. This means that only the language model and search space control word order. To compensate for this deficiency, the scheme uses a phrase constraint that limits switching between hypotheses. This constraint is included in the baseline system. With the match features, we drop this hard constraint on word order, finding better results.

6 Experiments

We use translations from the recent NIST Open MT 2009 (MT09) (Peterson et al., 2009a) and Fourth Workshop on Statistical Machine Translation (WMT) (Callison-Burch et al., 2009) evaluations. Results are reported for translations into English from Arabic and Urdu for MT09 and from Czech, German, Spanish, and French for WMT. Despite showing improvement of 1 BLEU point in translations from Hungarian, we elected to exclude this

language pair because automatic metrics perform poorly on this data. For example, the worst system according to BLEU by a significant margin was the best system according to human judges (Callison-Burch et al., 2009). The organizers of the following WMT also dropped Hungarian.

Official tuning and evaluation sets are used, except for MT09 Arabic-English where only unsequestered portions are used for evaluation. Language model training data for WMT is constrained to the provided English from monolingual and French-English corpora. There was no constrained informal system combination track for MT09 so we use a model trained on the Gigaword (Graff, 2003) corpus. Scores are reported using uncased BLEU (Papineni et al., 2002) from `mteval-13a.pl`, uncased TER (Snover et al., 2006) 0.7.25, and METEOR (Lavie and Denkowski, 2010) 1.0 with Adequacy-Fluency parameters.

For each source language, we selected a few subsets of systems to combine and picked the set that combined best on tuning data. Performance is surprisingly good on Arabic and competitive with top MT09 combinations. On French and Spanish, Google scored much higher than did other systems. Like Leusch et al. (2009b), we show no gain over Google on these source languages.

Most system combination schemes showed larger gains in MT09 than in WMT. In addition to different language pairs, one possible explanation is that MT09 has four references while WMT has one reference. Gains remained when scoring against one reference. Tuning towards multiple references conceivably increases individual system diversity, thereby increasing system combination's effectiveness. This is difficult to measure given only system outputs.

6.1 Maximum Match Length

The match features count only matches up to length l. Here, we ask what value of l is appropriate. Table 1 shows results for each source language for values of l ranging from 1 to 4.

In each scenario, using both unigram and bigram match ($l = 2$) features significantly outperforms using only unigram matches ($l = 1$), in one case by 6.7 BLEU points. This shows the significant role of match features in determining word order, without which the combination would be dependent on the language model and minimal guarantees from the search space. In particular, the search space permits hypotheses to switch systems at any point; the bigram features favor continuity across switches.

The effect of trigram and quadgram matches is mixed. On Arabic, improvement is significant and consistent where nine systems are combined. It decreases on Urdu with seven systems and vanishes for WMT with three to six systems combined. Fewer systems leave less opportunity for discrimination beyond unigrams and bigrams.

In	l	BLE	TER	MET
ar	1	51.2	41.5	73.4
	2	57.9	37.3	**76.9**
	3	58.0	37.2	76.8
	4	**58.6**	**37.0**	**76.9**
	i	*51.9*	*40.5*	*74.0*
cz	1	21.2	61.1	55.9
	2	21.7	60.6	55.9
	3	**21.8**	60.5	**56.0**
	4	**21.8**	60.9	55.9
	b	*21.7*	*60.8*	*54.8*
	i	*21.2*	***59.6***	*55.2*
es	1	27.7	54.7	61.8
	2	**28.9**	53.6	**62.2**
	3	28.7	53.6	**62.2**
	4	28.8	53.6	62.1
	b	*28.3*	*53.6*	*60.4*
	i	*28.7*	***53.4***	*62.0*

In	l	BLE	TER	MET
ur	1	31.4	57.2	60.1
	2	**34.7**	55.4	**62.3**
	3	34.5	**54.7**	61.7
	4	**34.7**	55.5	61.8
	i	*32.9*	*56.2*	*60.5*
de	1	21.4	60.3	57.3
	2	**23.8**	58.7	**58.7**
	3	23.7	**58.6**	58.5
	4	23.5	59.0	58.3
	b	*22.3*	*59.1*	*55.8*
	i	*21.3*	*60.8*	*57.0*
fr	1	29.5	52.6	62.4
	2	31.4	52.0	**63.3**
	3	**31.6**	52.7	**63.3**
	4	31.5	52.7	**63.3**
	b	*30.0*	*53.3*	*60.9*
	i	*31.1*	***51.4***	*62.8*

Table 1: Performance on test data by maximum match length l. For comparison, the baseline (b) cmu-combo from WMT 2009 is shown as well as the best individual system (i). Only exact alignments are counted as matches.

6.2 Importance of System Weights

Here, we compare tuned system weights with uniform system weights (Hildebrand and Vogel, 2009; Leusch et al., 2009a). We introduce hyper parameter t, tune system-level n-gram weights for $n \leq t$, and use uniform weights for $n > t$. Uniform weight is accomplished by replacing per-system n-gram count features with their sum.

Table 2 shows results on unigram and bigram features ($l = 2$). Tuning unigram weights improves performance in each scenario. With tuned bigram

In	t	BLE	TER	MET
ar	0	56.4	38.2	75.7
	1	57.0	37.9	75.9
	2	**57.9**	**37.3**	**76.9**
	i	*51.9*	*40.5*	*74.0*
cz	0	21.3	60.9	55.2
	1	21.5	60.7	**55.9**
	2	**21.7**	60.6	**55.9**
	b	***21.7***	*60.8*	*54.8*
	i	*21.2*	***59.6***	*55.2*
es	0	27.5	54.6	60.8
	1	28.7	53.7	62.1
	2	**28.9**	53.6	**62.2**
	b	*28.3*	*53.6*	*60.4*
	i	*28.7*	***53.4***	*62.0*

In	t	BLE	TER	MET
ur	0	33.2	56.3	61.6
	1	33.4	56.2	61.7
	2	**34.7**	**55.4**	**62.3**
	i	*32.9*	*56.2*	*60.5*
de	0	23.4	59.0	58.0
	1	23.7	**58.7**	58.6
	2	**23.8**	**58.7**	**58.7**
	b	*22.3*	*59.1*	*55.8*
	i	*21.3*	*60.8*	*57.0*
fr	0	29.3	53.7	62.2
	1	31.2	52.0	63.1
	2	**31.4**	52.0	**63.3**
	b	*30.0*	*53.3*	*60.9*
	i	*31.1*	***51.4***	*62.8*

Table 2: Impact of tuning system weights. Unigram and bigram matches are considered ($l = 2$); $t = 0$ has uniform weights, $t = 1$ tunes unigram weights, and $t = 2$ tunes bigram weights as well. For comparison, the baseline (b) cmu-combo from WMT 2009 is shown as well as the best individual system by BLEU (i). Only exact alignments are counted as matches.

weights, Arabic and Urdu show significant improvement; the others are within tolerance of retuning and length effects. Improvement from tuning unigram and bigram weights shows that systems differ in quality of lexical choice and word order, respectively.

6.3 Match Tolerance

As mentioned in Section 5, there are four types of alignments in decreasing order of priority: exact, stems, synonyms, and unigram paraphrases. We ask which alignments to use for the match features; the search space uses all alignments for its purposes. Since alignments are prioritized, we considered the four options ranging from using only exact alignments to using all alignments. In practice, performance meaningfully changes only after paraphrases are added. Therefore, we consider using exact alignments or using all alignments, shown in Table 3.

For French and German, higher BLEU scores result from counting exact alignments. French is characterized by Google scoring 4.2 BLEU higher than the second place system, with correspondingly high match feature weight. Counting all alignments awards this high weight to aligned word substitutions from weaker systems; counting exact alignments does not. Since BLEU cares about exact matches to the reference, it shows these differences the most.

When all alignments are counted, additional votes are collected on word inclusion and order. However, any gain in automatic metrics is minimal. METEOR is mostly apathetic to substitutions within its own alignments, so these scores are not expected to change much. For BLEU and TER, we wonder if gains from additional votes are offset by losses from lexical choice.

Given arguments for counting exact or all alignments, we now try both sets of features simultaneously. Since match feature weights are non-negative, this amounts to giving exact matches a tunable bonus. In experiments matching unigrams and bigrams ($l = 2$), performance usually mirrored that of the better performing set in isolation. However Arabic performance with simultaneous features is 58.55 BLEU, 36.86 TER, and 76.91 METEOR, which improves over results in Table 3. This is our best result with improvements of 6.67 BLEU, -3.68 TER, and 2.96 METEOR over the top individual system. Direct comparison to specific NIST systems is discouraged; results are in Peterson et al. (2009a).

In	a	BLE	TER	MET
ar	E	**57.9**	**37.3**	76.9
	A	**57.9**	**37.3**	**77.0**
	i	*51.9*	*40.5*	*74.0*
cz	E	21.7	60.6	55.9
	A	**21.8**	60.5	**56.1**
	b	*21.7*	*60.8*	*54.8*
	i	*21.2*	***59.6***	*55.2*
es	E	**28.9**	53.6	**62.2**
	A	28.7	53.7	**62.2**
	b	*28.3*	*53.6*	*60.4*
	i	*28.7*	***53.4***	*62.0*

In	a	BLE	TER	MET
ur	E	**34.7**	55.4	**62.3**
	A	34.5	**55.3**	**62.3**
	i	*32.9*	*56.2*	*60.5*
de	E	**23.8**	**58.7**	**58.7**
	A	23.2	59.0	**58.7**
	b	*22.3*	*59.1*	*55.8*
	i	*21.3*	*60.8*	*57.0*
fr	E	**31.4**	52.0	**63.3**
	A	30.6	52.8	63.2
	b	*30.0*	*53.3*	*60.9*
	i	*31.1*	***51.4***	*62.8*

Table 3: Combination performance with only exact (E) or all (A) alignments counted. For comparison, the baseline (b) cmu-combo from WMT 2009 is shown as well as the best individual system (i). The best result for each language and metric is bold.

6.4 Tuned Weights

In the previous experiments, we examined the tuned feature weights only by their impact on evaluation scores. It is also instructive to look at the weights themselves. Table 4 shows the weights for our best result, which combines nine Arabic systems. There are two copies of the match features: one that counts exact matches and another that counts any match identified by METEOR. Each copy considers both unigram and bigram matches ($l = 2$). Each of the nine systems therefore has four weights corresponding to four features: exact unigrams, exact bigrams, approximate unigrams, and approximate bigrams. Within each system, the features are highly correlated and the relative weight of highly correlated features is mostly arbitrary. Nonetheless, some broader trends appear.

The system weights are not a monotone function of BLEU. While the top system has generally high weight and the bottom system has generally low weight, system 16 in the middle has lower weights in each category than do those below it. We attribute this to system correlations; it is possible that system 16 and another system used the same decoder. Weighting systems by BLEU (Zhao and Jiang, 2009), a secondary method in Rosti et al. (2009), fails to account for correlations between systems so two strong but very similar systems would receive too much weight.

The ratio between unigram and bigram weights is not consistent. For system 14, unigram weights sum to 0.282 while bigram weights sum to 1.448, for a ratio of 0.195. System 7 has an analogous ratio of 1.752 while for system 8 it is 2.230. In conjunction with our previous experiment that found system-level bigram weights matter for Arabic, this suggests that separate system weights for unigrams and bigrams are more appropriate.

#	BLE	Exact Matches		All Matches	
		Unigram	Bigram	Unigram	Bigram
17	51.7	1.222	1.188	1.334	6.252
8	51.5	0.050	1.098	0.754	1.238
14	50.3	0.244	1.056	0.038	0.392
6	49.4	1.032	0.539	1.073	1.363
16	49.4	**0.032**	**0.698**	**0.658**	**0.246**
2	49.3	1.186	0.863	1.084	0.878
7	49.2	1.190	0.297	1.315	1.133
3	47.9	1.054	0.617	0.557	2.616
1	47.4	0.098	0.316	0.747	0.063

Table 4: Systems used in the top performing Arabic-English combination. Each system lists the anonymous system number assigned by NIST, uncased BLEU on the tuning set, and weights on that system's match features. There are two copies of the match features, one for exact alignments and another for all alignments. Scale of feature weights is arbitrary. While the top system has generally high weights and the bottom has generally low weights, there is no consistent pattern as a function of score. We draw particular attention to system 16 in **bold**, which has low weights across the board despite its position in the middle by BLEU score. The weights were tuned toward BLEU.

7 Conclusion

Our features address three core system combination problems: lexical choice, word order, and system weighting. We accomplish this by jointly tuning weights on the cross product of n-grams and systems, resulting in significant improvement on several combination tasks. This improvement comes from bigram matches, system-level weights, and in some cases flexible matching. These aspects are all variations on the fact that systems make different errors, which is system combination's raison d'être. Some or all of the aspects are missing in features used by other combination schemes. Our features are portable to these schemes, where we advocate their use.

Acknowledgments

This work was supported in part by the DARPA GALE program and by a NSF Graduate Research Fellowship.

References

Necip Fazil Ayan, Jing Zheng, and Wen Wang. 2008. Improving alignments for better confusion networks for combining machine translation systems. In *Proceedings 22nd International Conference on Computational Linguistics (COLING'2008)*, Manchester, UK, August.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. 2009. Findings of the 2009 Workshop on Statistical Machine Translation. In

Proceedings of the Fourth Workshop on Statistical Machine Translation, pages 1–28, Athens, Greece, March. Association for Computational Linguistics.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of EMNLP*, pages 224–233.

John DeNero, Shankar Kumar, Ciprian Chelba, and Franz Och. 2010. Model combination for machine translation. In *Proceedings NAACL-HLT*, Los Angeles, CA, June.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

David Graff. 2003. English gigaword. LDC2003T05.

Xiaodong He and Kristina Toutanova. 2009. Joint optimization for machine translation system combination. In *EMNLP*, August.

Kenneth Heafield, Greg Hanneman, and Alon Lavie. 2009. Machine translation system combination with flexible word ordering. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 56–60, Athens, Greece, March. Association for Computational Linguistics.

Almut Silja Hildebrand and Stephan Vogel. 2009. CMU system combination for WMT'09. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 47–50, Athens, Greece, March. Association for Computational Linguistics.

Damianos Karakos, Jason Eisner, Sanjeev Khudanpur, and Markus Dreyer. 2008. Machine translation system combination using ITG-based alignments. In *Proceedings ACL-08: HLT, Short Papers (Companion Volume)*, pages 81–84.

Damianos Karakos. 2009. The JHU system combination scheme. In *NIST Open Machine Translation Evaluation Workshop*, Ottawa, Canada, September.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, Prague, Czech Republic, June.

Shankar Kumar and William Byrne. 2004. Minimum bayes-risk decoding for statistical machine translation. In *Proceedings NAACL-HLT*, Boston, USA.

Alon Lavie and Michael Denkowski. 2010. The METEOR metric for automatic evaluation of machine translation. *MT Journal*.

Gregor Leusch, Saša Hasan, Saab Mansour, Matthias Huck, and Hermann Ney. 2009a. RWTH's system combination for the NIST 2009 MT ISC evaluation. In *NIST Open Machine Translation Evaluation Workshop*, Ottawa, Canada, September.

Gregor Leusch, Evgeny Matusov, and Hermann Ney. 2009b. The RWTH system combination system for WMT 2009. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 51–55, Athens, Greece, March. Association for Computational Linguistics.

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

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

Kay Peterson, Mark Przybocki, and Sébastien Bronsart. 2009a. NIST 2009 open machine translation evaluation (MT09) official release of results. http://www.itl.nist.gov/iad/mig/tests/mt/2009/.

Kay Peterson, Mark Przybocki, and Sébastien Bronsart. 2009b. NIST open machine translation evaluation 2009 human assessments. In *NIST Open Machine Translation 2009 Evaluation Workshop*, Ottawa, Canada, August.

Martin Porter. 2001. Snowball: A language for stemming algorithms. http://snowball.tartarus.org/.

Antti-Veikko I. Rosti, Spyros Matsoukas, and Richard Schwartz. 2007. Improved word-level system combination for machine translation. In *Proceedings 45th Annual Meeting of the Association of Computational Linguistics*, pages 312–319, Prague, Czech Republic, June.

Antti-Veikko I. Rosti, Bing Zhang, Spyros Matsoukas, and Richard Schwartz. 2008. Incremental hypothesis alignment for building confusion networks with application to machine translation system combination. In *Proceedings Third Workshop on Statistical Machine Translation*, pages 183–186.

Antti-Veikko Rosti, Spyros Matsoukas, and Bing Zhang. 2009. BBN A2E informal system combination submission. In *NIST Open Machine Translation Evaluation Workshop*, Ottawa, Canada, September.

K.C. Sim, W.J. Byrne, H. Sahbi M.J.F. Gales, and P.C. Woodland. 2007. Consensus network decoding for statistical machine translation system combination. In *Proceedings IEEE International Conference on Acoustics Speech*.

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

Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2008. TERp system description. In *Proceedings NIST Metrics MATR 2008*.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of the Seventh International Conference on Spoken Language Processing*, pages 901–904.

Richard Zens and Hermann Ney. 2006. N -gram posterior probabilities for statistical machine translation. In *Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL): Proceedings of the Workshop on Statistical Machine Translation*.

Yong Zhao and Xiaodong He. 2009. Using n-gram based features for machine translation system combination. In *Proceedings NAACL-HLT*, May.

Tiejun Zhao and Hongfei Jiang. 2009. The HIT-LTRC MT systems for NIST open machine translation 2009 evaluation. In *NIST Open Machine Translation Evaluation Workshop*, Ottawa, Canada, September.