CMU Stat-Xfer Group Informal System Combination

Kenneth Heafield and Alon Lavie

This document covers informal system combination in Urdu to English.

1 Site Affiliation

Carnegie Mellon Stat-Xfer Group (cmu-statxfer).

2 Contact Information

Kenneth Heafield <kheafiel@cs.cmu.edu> and Alon Lavie <alavie@cs.cmu.edu>

3 Submissions

- CMU-Stat-Xfer_u2e_isc_primary
- CMU-Stat-Xfer_u2e_isc_contrast1
- CMU-Stat-Xfer_u2e_isc_contrast2
- CMU-Stat-Xfer_u2e_isc_contrast3

4 Primary System Specs

4.1 Single System vs. System Combination

Informal System Combination

4.2 Core MT Engine Algorithmic Approach

The top five individual systems by BLEU score on the tuning set were used. We found that using more or fewer systems decreases the BLEU score of combined translations. The anonymized numbers of these systems, in decreasing order of uncased IBM BLEU on the tuning set, are 09, 08, 04, 05, and 07. These systems were combined using a method based on our previous work [3].

Each individual system provides a one-best translation. Untranslated words, identified by script, are removed. These five segments are aligned in pairs using the unigram alignment model of METEOR [1]. Briefly, the model aligns words exactly, by stemming, and by using the WordNet [2] synonymy database. Ambiguous alignments are resolved by minimizing crossing. The resulting alignments are precise but suffer in recall. A final step fills gaps by matching on part of speech where nearby words align.

The decoder starts with a single empty hypothesis. This branches into five hypotheses, one for each system, containing the first word from the system. Each hypothesis marks its respective word, and those aligned with it, as used. Each of the five resulting hypotheses branches into another five hypotheses by appending the next unused word from a system. When a system runs out of words, the hypothesis is recorded as a complete candidate sentence. It also continues to be extended using the remaining systems.

Systems often produce divergent and variable-length output, for which unigram alignment is incomplete. When one system's output is used for part of the sentence, unaligned words from other systems remain

unused. This leads to duplicated output when the lingering words are used later. A heuristic identifies these words and marks them as used. It does so by examining the position of the next word from each hypothesis. If the next word from a system lags another by more than four words then it is marked as used. The number four is a hyperparameter and was found to be best as described later under tuning.

Hypotheses are scored using several features. The first set of features measures how well the hypothesis is supported by the underlying systems. For each system, we count the number of hypothesis unigrams, bigrams, trigrams, and quadgrams supported by the system. Aligned words count as well, so these features are unchanged if the output hypothesis substitutes a synonym. With the five systems used, this is 20 features. With this many features, MERT is unstable and prone to overfitting, especially on the small number of sentences provided for tuning. Since the quadgram matches are rare, we also tried summing the five individual system quadgram counts to form one quadgram feature. This has the effect of forcing the quadgram features to have equal weight for each system. Similarly, we experimented with summing the trigram counts and forming one feature. The primary submission uses this configuration: separate unigram and bigram counts for each of the five systems and summed trigram and quadgram counts, for 12 features.

We also use two language model features: the log probability and length of match found. These come from an SRI language model with Kneser-Ney smoothing which was trained using GALE English corpora afp, ahram, apw, bbc, Bi-sakhr, cna, guardian, gulfnews, indian_src, ltw, nyt, people, taipei, US_src, xin, and yemen_times. Interpolation weights were tuned using the full NIST MT03 English set.

Length plays a role in hypothesis scoring. Like Moses [4], we have an explicit length feature. To first order, the length feature compensates for the impact of length on other feature values.

The aforementioned features are combined using a linear model. Model weights are tuned using MERT, which is challenged by the relatively large number of features. To encourage MERT to explore more model weights, we apply simulated annealing. In earlier iterations, the decoder randomly perturbs the model weights on a per-sentence basis. Specifically, each perturbed feature weight w' is independently sampled uniformly from the interval $[(1-s)w, (1+s)w]$ where w is the weight provided by MERT. The parameter s starts at 0.875 and decreases arithmetically by 0.125 for the first 8 iterations. The choice of these numbers is rather arbitrary. Another 12 iterations had $s = 0$ for no annealing and a total of 20 iterations. The number of iterations here is larger than typically used. This allows the simulated annealing to proceed slowly and compensates for time spent exploring.

This system has a number of hyperparameters in the sense that they are not trained by MERT. We tried 63 variants, each of which was tuned and evaluated on the tuning set. We tried various sets of systems, counting quadgrams and trigrams separately or summing, the two different types of position measurement, and various tolerances based on that measurement. We also tried traditional MERT versus MERT with simulated annealing. The system with highest uncased BLEU score was submitted as primary.

4.3 Critical Additional Features and Tools Used

A tokenizer from IBM intended for GALE was used. We used the METEOR metric [1] for alignments and evaluation. A Moses [4] recaser trained on constrained data was also used for final output.

4.4 Significant Data Pre/Post-Processing

Out-of-vocabulary words were filtered before running the combination. A simple pattern-based detokenizer was used after recasing.

4.5 Other Data Used (outside the prescribed LDC training data)

Our language model used GALE English corpora afp, ahram, apw, bbc, Bi-sakhr, cna, guardian, gulfnews, indian_src, ltw, nyt, people, taipei, US_src, xin, and yemen_times. In addition, MT03 was used to tune weights.

5 Key Difference In Contrastive Systems

Table 1 summarizes the submissions. Recall that the primary system relies on the length feature to correct for impact of length on the other features. In the contrastive systems, we kept the length feature but the other features were divided by length, thus averaging them over sentence length. This is the only difference between the primary and contrast1 submissions. The contrast2 submission differs from contrast1 in that contrast2 has all 20 support features rather than summing trigram and quadgram counts into one feature. Finally contrast3 differs more significantly. It combines the top seven systems (the top five plus 06 and 01) instead of the top five. Further, it uses a lag distance of 5 instead of 4 because this performed better on that set of systems.

Submission	**Separate Counts**	**Systems**	**Length Normalize**	**Lag Distance**	**BLEU**
primary	2	5	No	4	34.58
contrast1	2	5	Yes	4	34.34
contrast2	4	5	Yes	4	34.03
contrast3	4	7	Yes	5	34.19

Table 1: Differences between submissions. The column for separate counts indicates that n-gram matches up to the given length were reported as features on a per-system basis. The remaining n-gram counts up to length 4 were summed and reported as separate features. Case insensitive BLEU computed by `mteval-v13a.pl` is reported on the tuning set.

References

[1] Banerjee, S., and Lavie, A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proc. ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization* (2005), pp. 65–72.

[2] Fellbaum, C. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[3] Heafield, K., Hanneman, G., and Lavie, A. Machine translation system combination with flexible word ordering. In *Proc. Workshop on Machine Translation* (2009).

[4] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)* (Prague, Czech Republic, June 2007).