Positive Diversity Tuning for Machine Translation System Combination

Daniel Cer, Christopher D. Manning and Daniel Jurafsky
Stanford University
Stanford, CA 94305, USA
{danielcer,manning,jurafsky}@stanford.edu

Abstract

We present Positive Diversity Tuning, a new method for tuning machine translation models specifically for improved performance during system combination. System combination gains are often limited by the fact that the translations produced by the different component systems are too similar to each other. We propose a method for reducing excess cross-system similarity by optimizing a joint objective that simultaneously rewards models for producing translations that are similar to reference translations, while also punishing them for translations that are too similar to those produced by other systems. The formulation of the Positive Diversity objective is easy to implement and allows for its quick integration with most machine translation tuning pipelines. We find that individual systems tuned on the *same data* to Positive Diversity can be even more diverse than systems built using different data sets, while still obtaining good BLEU scores. When these individual systems are used together for system combination, our approach allows for significant gains of 0.8 BLEU even when the combination is performed using a small number of otherwise identical individual systems.

1 Introduction

The best performing machine translation systems are typically not individual decoders but rather are ensembles of two or more systems whose output is then merged using system combination algorithms. Since combining multiple distinct equally good translation systems reliably produces gains over any one of the systems in isolation, it is widely used in situations where high quality is essential.

Exploiting system combination brings significant cost: Macherey and Och (2007) showed that successful system combination requires the construction of multiple systems that are simultaneously diverse and well-performing. If the systems are not distinct enough, they will bring very little value during system combination. However, if some of the systems produce diverse translations but achieve lower overall translation quality, their contributions risk being ignored during system combination.

Prior work has approached the need for diverse systems by using different system architectures, model components, system build parameters, decoder hyperparameters, as well as data selection and weighting (Macherey and Och, 2007; DeNero et al., 2010; Xiao et al., 2013). However, during tuning, each individual system is still just trained to maximize its own isolated performance on a tune set, or at best an error-driven reweighting of the tune set, without explicitly taking into account the diversity of the resulting translations. Such tuning does not encourage systems to rigorously explore model variations that achieve both good translation quality and diversity with respect to the other systems. It is reasonable to suspect that this results in individual systems that under exploit the amount of diversity possible, given the characteristics of the individual systems.

For better system combination, we propose building individual systems to attempt to simultaneously maximize the overall quality of the individual systems and the amount of diversity across systems. We operationalize this problem formulation by devising a new heuristic measure called Positive Diversity that estimates the potential usefulness of individual systems during system combination. We find that optimizing systems toward Positive Diversity leads to significant performance gains during system combination even when the combination is performed using a small number of

otherwise identical individual translation systems.

The remainder of this paper is organized as follows. Section 2 and 3 briefly review the tuning of individual machine translation systems and how system combination merges the output of multiple systems into an improved combined translation. Section 4 introduces our Positive Diversity measure. Section 5 introduces an algorithm for training a collection of translation systems toward Positive Diversity. Experiments are presented in sections 6 and 7. Sections 8 and 9 conclude with discussions of prior work and directions for future research.

2 Tuning Individual Translation Systems

Machine translation systems are tuned toward some measure of the correctness of the translations produced by the system according to one or more manually translated references. As shown in equation (1), this can be written as finding parameter values Θ that produce translations sys_Θ that in turn achieve a high score on some correctness measure:

$$\arg\max_{\Theta} \ \texttt{Correctness}(\texttt{ref[]}, \texttt{sys}_\Theta) \qquad (1)$$

The correctness measure that systems are typically tuned toward is BLEU (Papineni et al., 2002), which measures the fraction of the n-grams that are both present in the reference translations and the translations produced by a system. The BLEU score is computed as the geometric mean of the resulting n-gram precisions scaled by a brevity penalty.

The most widely used machine translation tuning algorithm, minimum error rate training (MERT) (Och, 2003), attempts to maximize the correctness objective directly. Popular alternatives such as pairwise ranking objective (PRO) (Hopkins and May, 2011), MIRA (Chiang et al., 2008), and RAMPION (Gimpel and Smith, 2012) use surrogate optimization objectives that indirectly attempt to maximize the correctness function by using it to select targets for training discriminative classification models. In practice, either optimizing correctness directly or optimizing a surrogate objective that uses correctness to choose optimization targets results in roughly equivalent translation performance (Cherry and Foster, 2012).

Even when individual systems are being built to be used in a larger combined system, they are still usually tuned to maximize their isolated individual system performance rather than to maximize the potential usefulness of their contribution during system combination.[1] To our knowledge, no effort has been made to explicitly tune toward criteria that attempts to simultaneously maximize the translation quality of individual systems and their mutual diversity. This is unfortunate since the most valuable component systems for system combination should not only obtain good translation performance, but also produce translations that are different from those produced by other systems.

3 System Combination

Similar to speech recognition's Recognizer Output Voting Error Reduction (ROVER) algorithm (Fiscus, 1997), machine translation system combination typically operates by aligning the translations produced by two or more individual translation systems and then using the alignments to construct a search space that allows new translations to be pieced together by picking and choosing parts of the material from the original translations (Bangalore et al., 2001; Matusov et al., 2006; Rosti et al., 2007a; Rosti et al., 2007b; Karakos et al., 2008; Heafield and Lavie, 2010a).[2] The alignment of the individual system translations can be performed using alignment driven evaluation metrics such as invWER, TERp, METEOR (Leusch et al., 2003; Snover et al., 2009; Denkowski and Lavie, 2011). The piecewise selection of material from the original translations is performed using the combination model's scoring features such as n-gram language models, confidence models over the individual systems, and consensus features that score a combined translation using n-grams matches to the individual system translations (Rosti et al., 2007b; Zhao and He, 2009; Heafield and Lavie, 2010b).

Both system confidence model features and n-gram consensus features score contributions based in part on how confident the system combination model is in each individual machine translation system. This means that little or no gains will typically be seen when combining a good system with poor performing systems even if the systems col-

[1] The exception being Xiao et al. (2013)'s work using boosting for error-driven reweighting of the tuning set

[2] Other system combination techniques exist such as candidate selection systems, whereby the combination model attempts to find the best single candidate produced by one of the translation engines (Paul et al., 2005; Nomoto, 2004; Zwarts and Dras, 2008), decoder chaining (Aikawa and Ruopp, 2009), re-decoding informed by the decoding paths taken by other systems (Huang and Papineni, 2007), and decoding model combination (DeNero et al., 2010).

Input : systems [], `tune()`, source, refs [], α, `EvalMetric()`, `SimMetric()`
Output: models []

```
// start with an empty set of translations from prior iterations
other_sys [] ← []
for i ← 1 to len(systems []) do
    // new Positive Diversity measure using prior translations
    PD_{α,i}() ← new PD(α, EvalMetric(), SimMetric(), refs [], other_sys [])
    // tune a new model to fit PD_{α,i}
    // e.g., using MERT, PRO, MIRA, RAMPION, etc.
    models [i] ← tune(systems [i], source, PD_{α,i}())
    // Save translations from tuned model_i for use during
    // the diversity computation for subsequent systems
    push(other_sys [], translate(systems [i], models [i], source))
end
return models []
```

Algorithm 1: Positive Diversity Tuning (PDT)

lectively produce very diverse translations.[3]

The requirement that the systems used for system combination be both of high quality and diverse can be and often is met by building several different systems using different system architectures, model components or tuning data. However, as will be shown in the next few sections, by explicitly optimizing an objective that targets both translation quality and diversity, it is possible to obtain meaningful system combination gains even using a single system architecture with identical model components and the same tuning set.

4 Positive Diversity

We propose Positive Diversity as a heuristic measurement of the value of potential contributions from an individual system to system combination. As given in equation (2), Positive Diversity is defined as the correctness of the translations produced by a system minus a penalty term that scores how similar the systems translations are with those produced by other systems:

$$PD_\alpha = \alpha \ \texttt{Correctness}(\texttt{ref[]}, \texttt{sys}_\Theta) - (1-\alpha) \ \texttt{Similarity}(\texttt{other\_sys[]}, \texttt{sys}_\Theta) \quad (2)$$

The hyperparameter α explicitly trades-off the preference for a well performing individual system with system combination diversity. Higher α values result in a Positive Diversity metric that mostly favors good quality translations. However, even for large α values, if two translations are of approximately the same quality, the Positive Diversity metric will prefer the one that is the most diverse given the translations being produced by other systems.

The `Correctness()` and `Similarity()` measures are any function that can score translations from a single system against other translations. This includes traditional machine translation evaluation metrics (e.g, BLEU, TER, METEOR) as well as any other measure of textual similarity.

For the remainder of this paper, we use BLEU to measure both correctness and the similarity of the translations produced by the individual systems. When tuning individual translation systems toward Positive Diversity, our task is then to maximize equation (3) rather than equation (1):

$$\arg\max_\Theta \ \alpha \ \texttt{BLEU}(\texttt{ref[]}, \texttt{sys}) - (1-\alpha) \ \texttt{BLEU}(\texttt{other\_sys[]}, \texttt{sys}) \quad (3)$$

Since this learning objective is simply the difference between two BLEU scores, it should be easy to integrate into most existing machine translation tuning pipelines that are already designed to optimize performance on translation evaluation metrics.

[3] The machine learning theory behind boosting suggests that it should be possible to combine a very large number of poor performing systems into a single good system. However, for machine translation, using a very large number of individual systems brings with it difficult computational challenges.

PDT Individual System Diversity										
System \ Iteration	1	2	3	4	5	6	7	8	9	10
$\alpha = 0.95$	36.6	32.0	19.0	13.6	11.9	8.2	15.9	8.7	7.3	2.3
$\alpha = 0.97$	32.9	21.7	17.7	10.4	2.7	7.4	2.3	7.3	2.1	2.9
$\alpha = 0.99$	23.9	13.1	7.9	2.3	3.2	2.6	2.2	1.5	3.4	0.7

Table 1: Diversity scores for PDT individual systems on BOLT dev12 dev. Individual systems are tuned to Positive Diversity on GALE dev10 web tune. A system's diversity score is measured as its $1.0-$BLEU score on the translations produced by PDT systems from earlier iterations. Higher scores mean more diversity.

Diversity of Baseline System vs. Individual PDT Systems Available at Iteration i											
PDT Systems \ Iteration	0	1	2	3	4	5	6	7	8	9	10
$\alpha = 0.95$	27.3	20.4	16.8	14.9	12.8	11.4	9.4	8.6	8.3	8.1	7.9
$\alpha = 0.97$	28.4	21.3	15.8	14.7	13.3	13.0	12.5	12.2	10.3	10.0	9.7
$\alpha = 0.99$	27.5	22.6	18.5	17.1	16.8	15.9	15.4	14.6	14.3	13.5	13.4

Table 2: Diversity scores of a baseline system tuned to BOLT dev12 tune, a different tuning set than what was used for the PDT individual systems. The baseline system diversity is scored against all of the PDT individual systems available at iteration i for a given α value and over translations of BOLT dev12 dev.

5 Tuning to Positive Diversity

To tune a collection of machine translation systems using Positive Diversity, we propose a staged process, whereby systems are tuned one-by-one to maximize equation (2) using the translations produced by previously trained systems to compute the diversity term, `Similarity(other_sys[], sys`$_\Theta$`)`.

As shown in Algorithm 1, Positive Diversity Tuning (PDT) takes as input: a list of machine translation systems, `systems[]`; a tuning procedure for training individual systems, `tune()`; a tuning data set with source and reference translations, `source` and `refs`; a hyperparameter α to adjust the trade-off between fitting the reference translations and diversity between the systems; and metrics to measure correctness and cross-system similarity, `Correctness()` and `Similarity()`.

The list of systems can contain any translation system that can be parameterized using `tune()`. This can be a heterogeneous collection of substantially different systems (e.g., phrase-based, hierarchical, syntactic, or tunable hybrid systems) or even multiple copies of a single machine translation system. In all cases, systems later in the list will be trained to produce translations that both fit the references and are encouraged to be distinct from the systems earlier in the list.

During each iteration, the system constructs a new Positive Diversity measure $\texttt{PD}_{\alpha,i}$ using the translations produced during prior iterations of training. This $\texttt{PD}_{\alpha,i}$ measure is then given to `tune()` as the the training criteria for $\texttt{model}_i$ of $\texttt{system}_i$. The function `tune()` is any algorithm that allows a translation system's performance to be fit to an evaluation metric. This includes both minimum error rate training algorithms (MERT) that attempt to directly optimize a system's performance on a metric, as well as other techniques such as Pairwaise Ranking Objective (PRO), MIRA, and RAMPION that optimize a surrogate loss based on the preferences of an evaluation metric.

After training a model for each system, the resulting model-system pairs can be combined using any arbitrary system combination strategy.

6 Experiments

Experiments are performed using a single phrase-based Chinese-to-English translation system, built with the Stanford Phrasal machine translation toolkit (Cer et al., 2010). The system was built using all of the parallel data available for Phase 2 of the DARPA BOLT program. The Chinese data was segmented to the Chinese TreeBank (CTB) standard using a maximum match word segmenter, trained on the output of a CRF segmenter (Xiang et al., 2013). The bitext was word aligned using the Berkeley aligner (Liang et al., 2006). Standard phrase-pair extraction heuris-

	BLEU scores from individual systems tuned during iteration i of PDT										
PDT System	0	1	2	3	4	5	6	7	8	9	10
$\alpha = 0.95$	16.2	16.0	15.7	15.9	16.1	16.1	15.9	15.4	16.1	15.9	16.2
$\alpha = 0.97$	16.4	15.8	15.8	15.9	16.0	16.2	16.1	16.2	16.2	16.4	16.1
$\alpha = 0.99$	16.3	16.1	16.2	15.9	16.3	16.4	16.4	16.3	16.4	16.5	16.3

Table 3: BLEU scores on BOLT dev12 dev achieved by the individual PDT systems tuned on GALE dev10 web tune. Scores report individual system performance before system combination.

tics were used to extract a phrase-table over word alignments symmetrized using grow-diag (Koehn et al., 2003). We made use of a hierarchical reordering model (Galley and Manning, 2008) as well as a 5-gram language model trained on the target side of the bi-text and smoothed using modified Kneeser-Ney (Chen and Goodman, 1996).

Individual PDT systems were tuned on the GALE dev10 web tune set using online-PRO (Green et al., 2013; Hopkins and May, 2011) to the Positive Diversity Tuning criterion.[4] The Multi-Engine Machine Translation (MEMT) package was used for system combination (Heafield and Lavie, 2010a). We used BOLT dev12 dev as a development test set to explore different α parameterizations of the Positive Diversity criteria.

7 Results

Table 1 illustrates the amount of diversity achieved by individual PDT systems on the BOLT dev12 dev evaluation set for α values 0.95, 0.97, and 0.99.[5] Using different tuning sets is one of the common strategies for producing diverse component systems for system combination. Thus, as a baseline, Table 2 gives the diversity of a system tuned to BLEU using a different tuning set, BOLT dev12 tune, with respect to the PDT systems available at each iteration. As in Table 1, the diversity computation is performed using translations of BOLT dev12 dev.

Like the cross-system diversity term in the formulation of Positive Diversity using BLEU in equation (3), we measure the diversity of translations produced by an individual system as the negative BLEU score of the translations with respect to the translations from systems built during prior iterations. For clarity of presentation, these diversity scores are reported as $1.0-$BLEU. Using $1.0-$BLEU to score cross-system diversity, means that the reported numbers can be roughly interpreted as the fraction of n-grams from the individual systems built during iteration i that have not been previously produced by other systems built during any iteration $< i$.[6]

In our experiments, we find that for $\alpha \leq 0.97$, during the first three iterations of PDT, *there is more diversity among the PDT systems tuned on a single data set (GALE dev10 web tune) than there is between systems tuned on different datasets (BOLT dev12 tune vs. GALE dev10 wb tune)*. This is significant since using different tuning sets is a common strategy for increasing diversity during system combination. These results suggest PDT is better at producing additional diversity than using different tuning sets. The PDT systems also achieve good coverage of the n-grams present in the baseline system that was tuned using different data. At iteration 10 and using $\alpha = 0.95$, the baseline systems receive a diversity score of *only 7.9%* when measured against the PDT systems.[7]

As PDT progresses, it becomes more difficult to tune systems to produce high quality translations that are substantially different from those already being produced by other systems. This is seen in the per iteration diversity scores, whereby during iteration 5, the individual PDT translation systems have a $1.0-$BLEU diversity score with prior systems ranging from 11.9%, when using an α value

[4] Preliminary experiments performed using MERT to train the individual systems produced similar results to those seen here. However, we switched to online-PRO since it dramatically reduced the amount time required to train each individual system. We expect similar results when using other tuning algorithms for the individual systems, such as MIRA or RAMPION.

[5] Due to time constraints, we were not able to try additional α values. Given that our results suggest the lowest α value from the ones we tried works best (i.e., $\alpha = 0.95$), it would be worth trying additional smaller α values such as 0.90

[6] This intuitive interpretation assumes a brevity penalty that is approximately 1.0.

[7] For this diversity score, the brevity penalty is 1.0, meaning the diversity score is based purely on the n-grams present in the baseline system that are not present in translations produced by one or more of the PDT systems

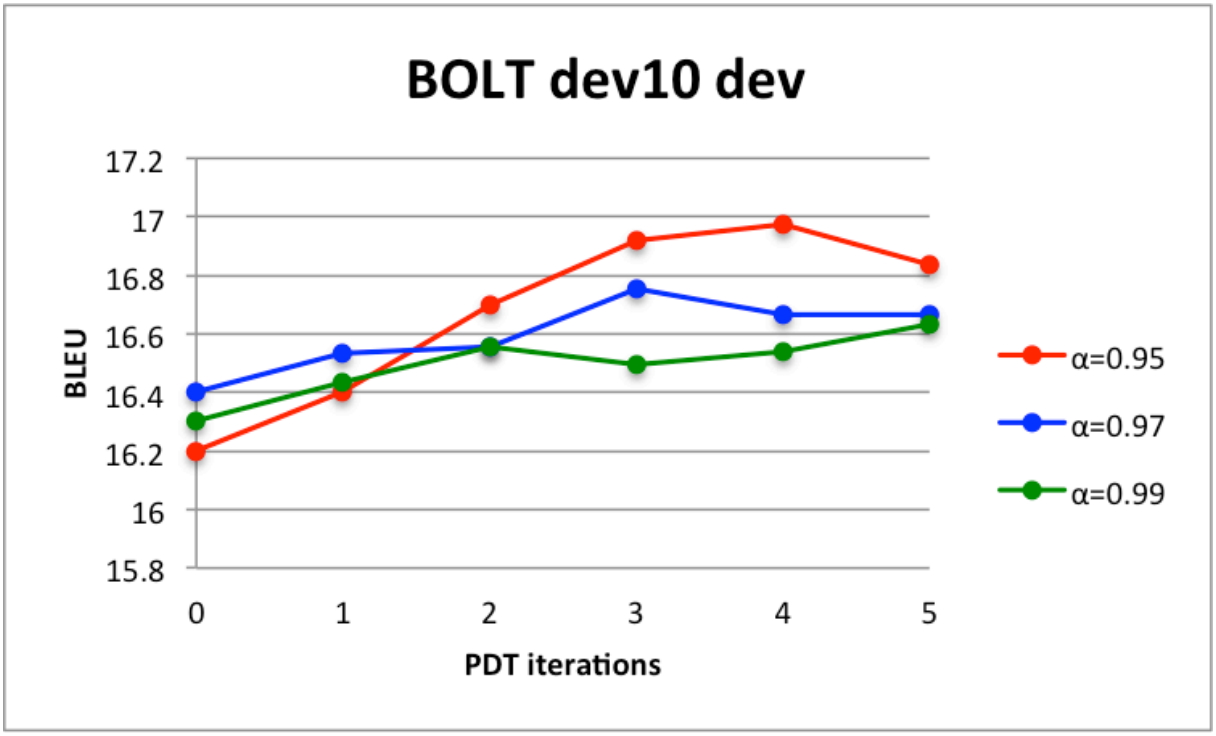

Figure 1: System combination BLEU score achieved using Positive Diversity Tuning with the α values 0.95, 0.97, and 0.99. Four iterations of PDT with $\alpha = 0.95$ results in a 0.8 BLEU gain over the initial BLEU tuned system. We only examine combinations of up to 6 systems (i.e., iterations 0-5), as the time required to tune MEMT increases dramatically as additional systems are added.

of 0.95, to 3.2% when using an α value of 0.99. A diversity score of 3.2% when using $\alpha = 0.99$ suggests that by iteration 5, very high α values put insufficient pressure on learning to find models that produce diverse translations. When using an α of 0.95, a sizable amount of diversity still exists across the systems translations all the way to iteration 7. By iteration 10, only a small amount of additional diversity is contributed by each additional system for all of the alpha values ($< 3\%$).[8]

Table 3 shows the BLEU scores obtained on the BOLT dev12 dev evaluation set by the *individual systems* tuned during each iteration of PDT. The 0^{th} iteration for each α value has an empty set of translations for the diversity term. This means the resulting systems are effectively tuned to just maximize BLEU. Differences in system performance during this iteration are only due to differences in the random seeds used during training. Starting at iteration 1, the individual systems are optimized to produce translations that both score well on BLEU and are diverse from the systems produced during prior iterations. It is interesting to note that the systems trained during these subsequent iterations obtain BLEU scores that are usually competitive with those obtained by the iteration 0 systems. Taken together with the diversity scores in Table 1, this strongly suggests that PDT is succeeding at increasing diversity while still producing high quality individual translation systems.

Figure 1 graphs the system combination BLEU score achieved by using varying numbers of Positive Diversity Tuned translation systems and different α values to trade-off translation quality with translation diversity. After running 4 iterations of PDT, the best configuration, $\alpha = 0.95$, achieves a BLEU score that is 0.8 BLEU higher than the corresponding BLEU trained iteration 0 system.[9]

From the graph, it appears that PDT performance initially increases as additional systems are added to the system combination and then later plateaus or even drops after too many systems are included. The combinations using PDT systems

[8] We speculate that if heterogeneous translation systems were used with PDT, it could be possible to run with higher α values and still obtain diverse translations after a large number of PDT iterations

[9] Recall that the iteration 0 system is effectively just tuned to maximize BLEU since we have an empty set of translations from other systems that are used to compute diversity

built with higher α values reach the point of diminishing returns faster than combinations using systems built with lower alpha values. For instance, $\alpha = 0.99$ plateaus on iteration 2, while $\alpha = 0.95$ peaks on iteration 4. It might be possible to identify the point at which additional systems will likely not be useful by using the diversity scores in Table 1. Scoring about 10% or less on the $1-$BLEU diversity measure, with respect to the other systems being used within the system combination, seems to suggest the individual system will not be very helpful to add into the combination.

8 Related Work

While the idea of encouraging diversity in individual systems that will be used for system combination has been proven effective in speech recognition and document summarization (Hinton, 2002; Breslin and Gales, 2007; Carbonell and Goldstein, 1998; Goldstein et al., 2000), there has only been a modest amount of prior work exploring such approaches for machine translation. Prior work within machine translation has investigated adapting machine learning techniques for building ensembles of classifiers to translation system tuning, encouraging diversity by varying both the hyperparameters and the data used to build the individual systems, and chaining together individual translation systems.

Xiao et al. (2013) explores using boosting to train an ensemble of machine translation systems. Following the standard Adaboost algorithm, each system was trained in sequence on an error-driven reweighting of the tuning set that focuses learning on the material that is the most problematic for the current ensemble. They found that using a single system to tune a large number of decoding models to different Adaboost guided weightings of the tuning data results in significant gains during system combination.

Macherey and Och (2007) investigated system combination using automatic generation of diverse individual systems. They programmatically generated variations of systems using different build and decoder hyperparameters such as choice of word-alignment algorithm, distortion limit, variations of model feature function weights, and the set of language models used. Then, in a process similar to forward feature selection, they constructed a combined system by iteratively adding the individual automatically generated system that produced the largest increase in quality when used in conjunction with the systems already selected for the combined system. They also explored producing variation by using different samplings of the the training data. The individual and combined systems produced by sampling the training data were inferior to systems that used all of the available data. However, the experiments facilitated insightful analysis on what properties an individual system must have in order to be useful during system combination. They found that in order to be useful within a combination, individual systems need to produce translations of similar quality to other individual systems within the system combination while also being as uncorrelated as possible from the other systems. The Positive Diversity Tuning method introduced in our work is an explicit attempt to build individual translation systems that meet this criteria, while being less computationally demanding than the diversity generating techniques explored by Macherey and Och (2007).

Aikawa and Ruopp (2009) investigated building machine translations systems specifically for use in sequential combination with other systems. They constructed chains of systems whereby the output of one decoder is feed as input to the next decoder in the pipeline. The downstream systems are built and tuned to correct errors produced by the preceding system. In this approach, the downstream decoder acts as a machine learning based post editing system.

9 Conclusion

We have presented Positive Diversity as a new way of jointly measuring the quality and diversity of the contribution of individual machine translation systems to system combination. This method heuristically assesses the value of individual translation systems by measuring their similarity to the reference translations as well as their dissimilarity from the other systems being combined. We operationalize this metric by reusing existing techniques from machine translation evaluation to assess translation quality and the degree of similarity between systems. We also give a straightforward algorithm for training a collection of individual systems to optimize Positive Diversity. Our experimental results suggest that tuning to Positive Diversity leads to improved cross-system diversity and system combination performance even when combining otherwise identical machine translation

systems.

The Positive Diversity Tuning method explored in this work can be used to tune individual systems for any ensemble in which individual models can be fit to multiple extrinsic loss functions. Since Hall et al. (2011) demonstrated the general purpose application of multiple extrinsic loss functions to training structured prediction models, Positive Diversity Tuning could be broadly useful within natural language processing and for other machine learning tasks.

In future work within machine translation, it may prove fruitful to examine more sophisticated measures of dissimilarity. For example, one could imagine a metric that punishes instances of similar material in proportion to some measure of the expected diversity of the material. It might also be useful to explore joint rather than sequential training of the individual translation systems.

Acknowledgments

We thank the reviewers and the members of the Stanford NLP group for their helpful comments and suggestions. This work was supported by the Defense Advanced Research Projects Agency (DARPA) Broad Operational Language Translation (BOLT) program through IBM and a fellowship to one of the authors from the Center for Advanced Study in the Behavioral Sciences. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of DARPA or the US government.

References

Takako Aikawa and Achim Ruopp. 2009. Chained system: A linear combination of different types of statistical machine translation systems. In *Proceedings of MT Summit XII*.

S. Bangalore, G. Bordel, and Giuseppe Riccardi. 2001. Computing consensus translation from multiple machine translation systems. In *ASRU*.

C. Breslin and M. J F Gales. 2007. Complementary system generation using directed decision trees. In *ICASSP*.

Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*.

Daniel Cer, Michel Galley, Daniel Jurafsky, and Christopher D. Manning. 2010. Phrasal: A statistical machine translation toolkit for Exploring new model features. In *NAACL/HLT*.

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *ACL*.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *NAACL/HLT*.

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

John DeNero, Shankar Kumar, Ciprian Chelba, and Franz Och. 2010. Model combination for machine translation. In *NAACL/HLT*.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the EMNLP 2011 Workshop on Statistical Machine Translation*.

J.G. Fiscus. 1997. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *ASRU*.

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

Kevin Gimpel and Noah A. Smith. 2012. Structured ramp loss minimization for machine translation. In *NAACL/HLT*.

J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *ANLP/NAACL Workshop on Automatic Summarization*.

Spence Green, Sida Wang, Daniel Cer, and Christopher D. Manning. 2013. Fast and adaptive online training of feature-rich translation models. In *(to appear) ACL*.

Keith Hall, Ryan McDonald, and Slav Petrov. 2011. Training structured prediction models with extrinsic loss functions. In *Domain Adaptation Workshop at NIPS*.

Kenneth Heafield and Alon Lavie. 2010a. CMU multi-engine machine translation for WMT 2010. In *WMT*.

Kenneth Heafield and Alon Lavie. 2010b. Voting on n-grams for machine translation system combination. In *AMTA*.

Geoffrey E. Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800, August.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *EMNLP*.

Fei Huang and Kishore Papineni. 2007. Hierarchical system combination for machine translation. In *EMNLP-CoNLL*.

Damianos Karakos, Jason Eisner, Sanjeev Khudanpur, and Markus Dreyer. 2008. Machine translation system combination using ITG-based alignments. In *ACL/HLT*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL*.

Gregor Leusch, Nicola Ueffing, and Hermann Ney. 2003. A novel string-to-string distance measure with applications to machine translation evaluation. In *MT Summit*.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *NAACL/HLT*.

Wolfgang Macherey and Franz J. Och. 2007. An empirical study on computing consensus translations from multiple machine translation systems. In *EMNLP/CoNLL*.

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

Tadashi Nomoto. 2004. Multi-engine machine translation with voted language model. In *ACL*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Michael Paul, Takao Doi, Youngsook Hwang, Kenji Imamura, Hideo Okuma, and Eiichiro Sumita. 2005. Nobody is perfect: ATR's hybrid approach to spoken language translation. In *IWSLT*.

Antti-Veikko Rosti, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie Dorr. 2007a. Combining outputs from multiple machine translation systems. In *NAACL/HLT*.

Antti-Veikko Rosti, Spyros Matsoukas, and Richard Schwartz. 2007b. Improved word-level system combination for machine translation. In *ACL*.

Matthew Snover, Nitin Madnani, Bonnie J. Dorr, and Richard Schwartz. 2009. Fluency, adequacy, or HTER?: exploring different human judgments with a tunable MT metric. In *WMT*.

Bing Xiang, Xiaoqiang Luo, and Bowen Zhou. 2013. Enlisting the ghost: Modeling empty categories for machine translation. In *ACL*.

Tong Xiao, Jingbo Zhu, and Tongran Liu. 2013. Bagging and boosting statistical machine translation systems. *Artif. Intell.*, 195:496–527, February.

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

Simon Zwarts and Mark Dras. 2008. Choosing the right translation: A syntactically informed classification approach. In *CoLING*.