Discriminative Feature-Tied Mixture Modeling for Statistical Machine Translation

Bing Xiang and Abraham Ittycheriah
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
{bxiang,abei}@us.ibm.com

Abstract

In this paper we present a novel discriminative mixture model for statistical machine translation (SMT). We model the feature space with a log-linear combination of multiple mixture components. Each component contains a large set of features trained in a maximum-entropy framework. All features within the same mixture component are tied and share the same mixture weights, where the mixture weights are trained discriminatively to maximize the translation performance. This approach aims at bridging the gap between the maximum-likelihood training and the discriminative training for SMT. It is shown that the feature space can be partitioned in a variety of ways, such as based on feature types, word alignments, or domains, for various applications. The proposed approach improves the translation performance significantly on a large-scale Arabic-to-English MT task.

1 Introduction

Significant progress has been made in statistical machine translation (SMT) in recent years. Among all the proposed approaches, the phrase-based method (Koehn et al., 2003) has become the widely adopted one in SMT due to its capability of capturing local context information from adjacent words. There exists significant amount of work focused on the improvement of translation performance with better features. The feature set could be either small (at the order of 10), or large (up to millions). For example, the system described in (Koehn et al., 2003) is a widely known one using small number of features in a maximum-entropy (log-linear) model (Och and Ney, 2002). The features include phrase translation probabilities, lexical probabilities, number of phrases, and language model scores, etc. The feature weights are usually optimized with minimum error rate training (MERT) as in (Och, 2003).

Besides the MERT-based feature weight optimization, there exist other alternative discriminative training methods for MT, such as in (Tillmann and Zhang, 2006; Liang et al., 2006; Blunsom et al., 2008). However, scalability is a challenge for these approaches, where all possible translations of each training example need to be searched, which is computationally expensive.

In (Chiang et al., 2009), there are 11K syntactic features proposed for a hierarchical phrase-based system. The feature weights are trained with the Margin Infused Relaxed Algorithm (MIRA) efficiently on a forest of translations from a development set. Even though significant improvement has been obtained compared to the baseline that has small number of features, it is hard to apply the same approach to millions of features due to the data sparseness issue, since the development set is usually small.

In (Ittycheriah and Roukos, 2007), a maximum entropy (ME) model is proposed, which utilizes millions of features. All the feature weights are trained with a maximum-likelihood (ML) approach on the full training corpus. It achieves significantly better performance than a normal phrase-based system. However, the estimation of feature weights has no direct connection with the final translation perfor-

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 424–428,
Portland, Oregon, June 19-24, 2011.

mance.

In this paper, we propose a hybrid framework, a discriminative mixture model, to bridge the gap between the ML training and the discriminative training for SMT. In Section 2, we briefly review the ME baseline of this work. In Section 3, we introduce the discriminative mixture model that combines various types of features. In Section 4, we present experimental results on a large-scale Arabic-English MT task with focuses on feature combination, alignment combination, and domain adaptation, respectively. Section 5 concludes the paper.

2 Maximum-Entropy Model for MT

In this section we give a brief review of a special maximum-entropy (ME) model as introduced in (Ittycheriah and Roukos, 2007). The model has the following form,

$$p(\mathbf{t}, j|\mathbf{s}) = \frac{p_0(\mathbf{t}, j|\mathbf{s})}{Z(\mathbf{s})} exp \sum_i \lambda_i \phi_i(\mathbf{t}, j, \mathbf{s}), \quad (1)$$

where $\mathbf{s}$ is a source phrase, and $\mathbf{t}$ is a target phrase. j is the jump distance from the previously translated source word to the current source word. During training j can vary widely due to automatic word alignment in the parallel corpus. To limit the sparseness created by long jumps, j is capped to a window of source words (-5 to 5 words) around the last translated source word. Jumps outside the window are treated as being to the edge of the window. In Eq. (1), p_0 is a prior distribution, Z is a normalizing term, and $\phi_i(\mathbf{t}, j, \mathbf{s})$ are the features of the model, each being a binary question asked about the source, distortion, and target information. The feature weights λ_i can be estimated with the Improved Iterative Scaling (IIS) algorithm (Della Pietra et al., 1997), a maximum-likelihood-based approach.

3 Discriminative Mixture Model

3.1 Mixture Model

Now we introduce the discriminative mixture model. Suppose we partition the feature space into multiple clusters (details in Section 3.2). Let the probability of target phrase and jump given certain source phrase for cluster k be

$$p_k(\mathbf{t}, j|\mathbf{s}) = \frac{1}{Z_k(\mathbf{s})} exp \sum_i \lambda_{ki} \phi_{ki}(\mathbf{t}, j, \mathbf{s}), \quad (2)$$

where Z_k is a normalizing factor for cluster k.

We propose a log-linear mixture model as shown in Eq. (3).

$$p(\mathbf{t}, j|\mathbf{s}) = \frac{p_0(\mathbf{t}, j|\mathbf{s})}{Z(\mathbf{s})} \prod_k p_k(\mathbf{t}, j|\mathbf{s})^{w_k}. \quad (3)$$

It can be rewritten in the log domain as

$$\begin{aligned} \log p(\mathbf{t}, j|\mathbf{s}) &= \log \frac{p_0(\mathbf{t}, j|\mathbf{s})}{Z(\mathbf{s})} \\ &\quad + \sum_k w_k \log p_k(\mathbf{t}, j|\mathbf{s}) \\ &= \log \frac{p_0(\mathbf{t}, j|\mathbf{s})}{Z(\mathbf{s})} - \sum_k w_k \log Z_k(\mathbf{s}) \\ &\quad + \sum_k w_k \sum_i \lambda_{ki} \phi_{ki}(\mathbf{t}, j, \mathbf{s}). \quad (4) \end{aligned}$$

The individual feature weights λ_{ki} for the i-th feature in cluster k are estimated in the maximum-entropy framework as in the baseline model. However, the mixture weights w_k can be optimized directly towards the translation evaluation metric, such as BLEU (Papineni et al., 2002), along with other usual costs (e.g. language model scores) on a development set. Note that the number of mixture components is relatively small (less than 10) compared to millions of features in baseline. Hence the optimization can be conducted easily to generate reliable mixture weights for decoding with MERT (Och, 2003) or other optimization algorithms, such as the Simplex Armijo Downhill algorithm proposed in (Zhao and Chen, 2009).

3.2 Partition of Feature Space

Given the proposed mixture model, how to split the feature space into multiple regions becomes crucial. In order to surpass the baseline model, where all features can be viewed as existing in a single mixture component, the separated mixture components should be complementary to each other. In this work, we explore three different ways of partitions, based on either feature types, word alignment types, or the domain of training data.

In the feature-type-based partition, we split the ME features into 8 categories:

- F1: Lexical features that examine source word, target word and jump;

- F2: Lexical context features that examine source word, target word, the previous source word, the next source word and jump;
- F3: Lexical context features that examine source word, target word, the previous source word, the previous target word and jump;
- F4: Lexical context features that examine source word, target word, the previous or next source word and jump;
- F5: Segmentation features based on morphological analysis that examine source morphemes, target word and jump;
- F6: Part-of-speech (POS) features that examine the source and target POS tags and their neighbors, along with target word and jump;
- F7: Source parse tree features that collect the information from the parse labels of the source words and their siblings in the parse trees, along with target word and jump;
- F8: Coverage features that examine the coverage status of the source words to the left and to the right. They fire only if the left source is open (untranslated) or the right source is closed.

All the features falling in the same feature category/cluster are tied to each other to share the same mixture weights at the upper level as in Eq. (3).

Besides the feature-type-based clustering, we can also divide the feature space based on word alignment types, such as supervised alignment versus unsupervised alignment (to be described in the experiment section). For each type of word alignment, we build a mixture component with millions of ME features. On the task of domain adaptation, we can also split the training data based on their domain/resources, with each mixture component representing a specific domain.

4 Experiments

4.1 Data and Baseline

We conduct a set of experiments on an Arabic-to-English MT task. The training data includes the UN parallel corpus and LDC-released parallel corpora, with about 10M sentence pairs and 300M words in total (counted at the English side). For each sentence in the training, three types of word alignments are created: maximum entropy alignment (Ittycheriah and Roukos, 2005), GIZA++ alignment (Och and Ney, 2000), and HMM alignment (Vogel et al., 1996). Our tuning and test sets are extracted from the GALE DEV10 Newswire set, with no overlap between tuning and test. There are 1063 sentences (168 documents) in the tuning set, and 1089 sentences (168 documents) in the test set. Both sets have one reference translation for each sentence. Instead of using all the training data, we sample the training corpus based on the tuning/test set to train the systems more efficiently. In the end, about 1.5M sentence pairs are selected for the sampled training. A 5-gram language model is trained from the English Gigaword corpus and the English portion of the parallel corpus used in the translation model training. In this work, the decoding weights for both the baseline and the mixture model are tuned with the Simplex Armijo Downhill algorithm (Zhao and Chen, 2009) towards the maximum BLEU.

System	Features	BLEU
F1	685K	37.11
F2	5516K	38.43
F3	4457K	37.75
F4	3884K	37.56
F5	103K	36.03
F6	325K	37.89
F7	1584K	38.56
F8	1605K	37.49
Baseline	18159K	39.36
Mixture	18159K	39.97

Table 1: MT results with individual mixture component (F1 to F8), baseline, or mixture model.

4.2 Feature Combination

We first experiment with the feature-type-based clustering as described in Section 3.2. The translation results on the test set from the baseline and the mixture model are listed in Table 1. The MT performance is measured with the widely adopted BLEU metric. We also evaluate the systems that utilize only one of the mixture components (F1 to F8). The number of features used in each system is also

listed in the table. As we can see, when using all 18M features in the baseline model, without mixture weighting, the baseline achieved 3.3 points higher BLEU score than F5 (the worst component), and 0.8 higher BLEU score than F7 (the best component). With the log-linear mixture model, we obtained 0.6 gain compared to the baseline. Since there are exactly the same number of features in the baseline and mixture model, the better performance is due to two facts: separate training of the feature weights λ within each mixture component; the discriminative training of mixture weights w. The first one allows better parameter estimation given the number of features in each mixture component is much less than that in the baseline. The second factor connects the mixture weighting to the final translation performance directly. In the baseline, all feature weights are trained together solely under the maximum likelihood criterion, with no differentiation of the various types of features in terms of their contribution to the translation performance.

System	Features	BLEU
ME	5687K	39.04
GIZA	5716K	38.75
HMM	5589K	38.65
Baseline	18159K	39.36
Mixture	16992K	39.86

Table 2: MT results with different alignments, baseline, or mixture model.

4.3 Alignment Combination

In the baseline mentioned above, three types of word alignments are used (via corpus concatenation) for phrase extraction and feature training. Given the mixture model structure, we can apply it to an alignment combination problem. With the phrase table extracted from all the alignments, we train three feature mixture components, each on one type of alignments. Each mixture component contains millions of features from all feature types described in Section 3.2. Again, the mixture weights are optimized towards the maximum BLEU. The results are shown in Table 2. The baseline system only achieved 0.3 minor gain compared to extracting features from ME alignment only (note that phrases are from all the alignments). With the mixture model, we can achieve another 0.5 gain compared to the baseline, especially with less number of features. This presents a new way of doing alignment combination in the feature space instead of in the usual phrase space.

System	Features	BLEU
Newswire	8898K	38.82
Weblog	1990K	38.20
UN	4700K	38.21
Baseline	18159K	39.36
Mixture	15588K	39.81

Table 3: MT results with different training sub-corpora, baseline, or mixture model.

4.4 Domain Adaptation

Another popular task in SMT is domain adaptation (Foster et al., 2010). It tries to take advantage of any out-of-domain training data by combining them with the in-domain data in an appropriate way. In our sub-sampled training corpus, there exist three subsets: newswire (1M sentences), weblog (200K), and UN data (300K). We train three mixture components, each on one of the training subsets. All results are compared in Table 3. The baseline that was trained on all the data achieved 0.5 gain compared to using the newswire training data alone (understandably it is the best component given the newswire test data). Note that since the baseline is trained on sub-sampled training data, there is already certain domain adaptation effect involved. On top of that, the mixture model results in another 0.45 gain in BLEU. All the improvements in the mixture models above against the baseline are statistically significant with p-value < 0.0001 by using the confidence tool described in (Zhang and Vogel, 2004).

5 Conclusion

In this paper we presented a novel discriminative mixture model for bridging the gap between the maximum-likelihood training and the discriminative training in SMT. We partition the feature space into multiple regions. The features in each region are tied together to share the same mixture weights that are optimized towards the maximum BLEU scores. It was shown that the same model structure can be ef-

fectively applied to feature combination, alignment combination and domain adaptation. We also point out that it is straightforward to combine any of these three. For example, we can cluster the features based on both feature types and alignments. Further improvement may be achieved with other feature space partition approaches in the future.

Acknowledgments

We would like to acknowledge the support of DARPA under Grant HR0011-08-C-0110 for funding part of this work. The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

References

Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008. A discriminative latent variable model for statistical machine translation. In *Proceedings of ACL-08:HLT*.

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of NAACL-HLT*.

Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. 1997. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in satistical machine translation. In *Proceedings of EMNLP*.

Abraham Ittycheriah and Salim Roukos. 2005. A maximum entropy word aligner for arabic-english machine translation. In *Proceedings of HLT/EMNLP*, pages 89–96, October.

Abraham Ittycheriah and Salim Roukos. 2007. Direct translation model 2. In *Proceedings HLT/NAACL*, pages 57–64, April.

Philipp Koehn, Franz Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL/HLT*.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proceedings of ACL/COLING*, pages 761–768, Sydney, Australia.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of ACL*, pages 440–447, Hong Kong, China, October.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translations. In *Proceedings of ACL*, pages 295–302, Philadelphia, PA, July.

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 Weijing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Christoph Tillmann and Tong Zhang. 2006. A discriminative global training algorithm for statistical mt. In *Proceedings of ACL/COLING*, pages 721–728, Sydney, Australia.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of COLING*, pages 836–841.

Ying Zhang and Stephan Vogel. 2004. Measuring confidence intervals for the machine translation evaluation metrics. In *Proceedings of The 10th International Conference on Theoretical and Methodological Issues in Machine Translation*.

Bing Zhao and Shengyuan Chen. 2009. A simplex armijo downhill algorithm for optimizing statistical machine translation decoding parameters. In *Proceedings of NAACL-HLT*.