Applying Pairwise Ranked Optimisation to Improve the Interpolation of Translation Models

Barry Haddow
University of Edinburgh
Scotland
bhaddow@inf.ed.ac.uk

Abstract

In Statistical Machine Translation we often have to combine different sources of parallel training data to build a good system. One way of doing this is to build separate translation models from each data set and linearly interpolate them, and to date the main method for optimising the interpolation weights is to minimise the model perplexity on a heldout set. In this work, rather than optimising for this indirect measure, we directly optimise for BLEU on the tuning set and show improvements in average performance over two data sets and 8 language pairs.

1 Introduction

Statistical Machine Translation (SMT) requires large quantities of parallel training data in order to produce high quality translation systems. This training data, however, is often scarce and must be drawn from whatever sources are available. If these data sources differ systematically from each other, and/or from the test data, then the problem of combining these disparate data sets to create the best possible translation system is known as *domain adaptation*.

One approach to domain adaptation is to build separate models for each training domain, then weight them to create a system tuned to the test domain. In SMT, a successful approach to building domain specific language models is to build one from each corpus, then linearly interpolate them, choosing weights that minimise the perplexity on a suitable heldout set of in-domain data. This method has been applied by many authors (e.g. (Koehn and Schroeder, 2007)), and is implemented in popular language modelling tools like IRSTLM (Federico et al., 2008) and SRILM (Stolcke, 2002).

Similar interpolation techniques have been developed for translation model interpolation (Foster et al., 2010; Sennrich, 2012) for phrase-based systems but have not been as widely adopted, perhaps because the efficacy of the methods is not as clear-cut. In this previous work, the authors used standard phrase extraction heuristics to extract phrases from a heldout set of parallel sentences, then tuned the translation model (i.e. the phrase table) interpolation weights to minimise the perplexity of the interpolated model on this set of extracted phrases.

In this paper, we try to improve on this perplexity optimisation of phrase table interpolation weights by addressing two of its shortcomings. The first problem is that the perplexity is not well defined because of the differing coverage of the phrase tables, and their partial coverage of the phrases extracted from the heldout set. Secondly, perplexity may not correlate with the performance of the final SMT system.

So, instead of optimising the interpolation weights for the indirect goal of translation model perplexity, we optimise them directly for translation performance. We do this by incorporating these weights into SMT tuning using a modified version of Pairwise Ranked Optimisation (PRO) (Hopkins and May, 2011).

In experiments on two different domain adaptation problems and 8 language pairs, we show that our method achieves comparable or improved performance, when compared to the perplexity minimisation method. This is an encouraging result as it

Proceedings of NAACL-HLT 2013, pages 342–347,
Atlanta, Georgia, 9–14 June 2013.

shows that PRO can be adapted to optimise translation parameters other than those in the standard linear model.

2 Optimising Phrase Table Interpolation Weights

2.1 Previous Approaches

In the work of Foster and Kuhn (2007), linear interpolation weights were derived from different measures of distance between the training corpora, but this was not found to be successful. Optimising the weights to minimise perplexity, as described in the introduction, was found by later authors to be more useful (Foster et al., 2010; Sennrich, 2012), generally showing small improvements over the default approach of concatenating all training data.

An alternative approach is to use log-linear interpolation, so that the interpolation weights can be easily optimised in tuning (Koehn and Schroeder, 2007; Bertoldi and Federico, 2009; Banerjee et al., 2011). However, this effectively multiplies the probabilities across phrase tables, which does not seem appropriate, especially for phrases absent from 1 table.

2.2 Tuning SMT Systems

The standard SMT model scores translation hypotheses as a linear combination of features. The model score of a hypothesis e is then defined to be $\mathbf{w} \cdot \mathbf{h}(e, f, a)$ where $\mathbf{w}$ is a weight vector, and $\mathbf{h}(e, f, a)$ a vector of feature functions defined over source sentences (f), hypotheses, and their alignments (a). The weights are normally optimised (*tuned*) to maximise BLEU on a heldout set (the *tuning* set).

The most popular algorithm for this weight optimisation is the line-search based MERT (Och, 2003), but recently other algorithms that support more features, such as PRO (Hopkins and May, 2011) or MIRA-based algorithms (Watanabe et al., 2007; Chiang et al., 2008; Cherry and Foster, 2012), have been introduced. All these algorithms assume that the model score is a linear function of the parameters $\mathbf{w}$. However since the phrase table probabilities enter the score function in log form, if these probabilities are a linear interpolation, then the model score is not a linear function of the interpolation weights. We will show that PRO can be used to simultaneously optimise such non-linear parameters.

2.3 Pairwise Ranked Optimisation

PRO is a *batch* tuning algorithm in the sense that there is an outer loop which repeatedly decodes a small (1000-2000 sentence) tuning set and passes the n-best lists from this tuning set to the core algorithm (also known as the *inner loop*). The core algorithm samples pairs of hypotheses from the n-best lists (according to a specific procedure), and uses these samples to optimise the weight vector $\mathbf{w}$.

The core algorithm in PRO will now be explained in more detail. Suppose that the N sampled hypothesis pairs (x_i^α, x_i^β) are indexed by i and have corresponding feature vectors pairs $(\mathbf{h}_i^\alpha, \mathbf{h}_i^\beta)$. If the gain of a given hypothesis (we use smoothed sentence BLEU) is given by the function $g(x)$, then we define y_i by

$$y_i \equiv \text{sgn}(g(x_i^\alpha) - g(x_i^\beta)) \quad (1)$$

For weights $\mathbf{w}$, and hypothesis pair (x_i^α, x_i^β), the (model) score difference $\Delta s_i^\mathbf{w}$ is given by:

$$\Delta s_i^\mathbf{w} \equiv s^\mathbf{w}(x_i^\alpha) - s^\mathbf{w}(x_i^\beta) \equiv \mathbf{w} \cdot \left(\mathbf{h}_i^\alpha - \mathbf{h}_i^\beta\right) \quad (2)$$

Then the core PRO algorithm updates the weight vector to $\mathbf{w}^*$ by solving the following optimisation problem:

$$\mathbf{w}^* = \arg\max_\mathbf{w} \sum_{i=1}^{N} \log\left(\sigma\left(y_i \Delta s_i^\mathbf{w}\right)\right) \quad (3)$$

where $\sigma(x)$ is the standard sigmoid function. The derivative of the function can be computed easily, and the optimisation problem can be solved with standard numerical optimisation algorithms such as L-BFGS (Byrd et al., 1995). PRO is normally implemented by converting each sample to a training example for a 2 class maximum entropy classifier, with the feature values set to $\Delta\mathbf{h}_i$ and the responses set to the y_i, whereupon the log-likelihood is the objective given in Equation (3). As in maximum entropy modeling, it is usual to add a Gaussian prior to the objective (3) in PRO training.

2.4 Extending PRO for Mixture Models

We now show how to apply the PRO tuning algorithm of the previous subsection to simultaneously

optimise the weights of the translation system, and the interpolation weights.

In the standard phrase-based model, some of the features are derived from logs of phrase translation probabilities. If the phrase table is actually a linear interpolation of two (or more) phrase tables, then we can consider these features as also being functions of the interpolation weights. The interpolation weights then enter the score differences $\{\Delta s_i^{\mathbf{w}}\}$ via the phrase features, and we can jointly optimise the objective in Equation (3) for translation model weights and interpolation weights.

To make this more concrete, suppose that the feature vector consists of m phrase table features and $n - m$ other features[1]

$$\mathbf{h} \equiv (\log(p^1), \ldots, \log(p^m), h^{m+1}, \ldots h^n) \quad (4)$$

where each p^j is an interpolation of two probability distributions p_A^j and p_B^j. So, $p^j \equiv \lambda^j p_A^j + (1-\lambda^j) p_B^j$ with $0 \leq \lambda^j \leq 1$. Defining $\boldsymbol{\lambda} \equiv (\lambda^1 \ldots \lambda^m)$, the optimisation problem is then:

$$(\mathbf{w}^*, \boldsymbol{\lambda}^*) = \arg\max_{(\mathbf{w},\boldsymbol{\lambda})} \sum_{i=1}^{N} \log\left(\sigma\left(y_i \Delta s_i^{(\mathbf{w},\boldsymbol{\lambda})}\right)\right) \quad (5)$$

where the sum is over the sampled hypothesis pairs and the Δ indicates the difference between the model scores of the two hypotheses in the pair, as before. The model score $s_i^{(\mathbf{w},\boldsymbol{\lambda})}$ is given by

$$\sum_{j=1}^{m} \left(w^j \cdot \log\left(\lambda^j p_{Ai}^j + (1-\lambda^j) p_{Bi}^j)\right)\right) + \sum_{j=m+1}^{n} w^j h_i^j \quad (6)$$

where $\mathbf{w} \equiv (w^i \ldots w^n)$. A Gaussian regularisation term is added to the objective, as it was for PRO. By replacing the core algorithm of PRO with the optimisation above, the interpolation weights can be trained simultaneously with the other model weights.

Actually, the above explanation contains a simplification, in that it shows the phrase features interpolated at sentence level. In reality the phrase features are interpolated at the phrase level, then combined to give the sentence level feature value. This makes the definition of the objective more complex than that shown above, but still optimisable using bounded L-BFGS.

[1] Since the phrase penalty feature is a constant across phrase pairs it is not interpolated, and so is classed with the the "other" features. The lexical scores, although not actually probabilities, are interpolated.

3 Experiments

3.1 Corpus and Baselines

We ran experiments with data from the WMT shared tasks (Callison-Burch et al., 2007; Callison-Burch et al., 2012), as well as OpenSubtitles data[2] released by the OPUS project (Tiedemann, 2009).

The experiments targeted both the news-commentary (`nc`) and OpenSubtitles (`st`) domains, with `nc-devtest2007` and `nc-test2007` for tuning and testing in the `nc` domain, respectively, and corresponding 2000 sentence tuning and test sets selected from the `st` data. The news-commentary v7 corpus and a 200k sentence corpus selected from the remaining `st` data were used as in-domain training data for the respective domains, with europarl v7 (`ep`) used as out-of-domain training data in both cases. The language pairs we tested were the WMT language pairs for `nc` (English (en) to and from Spanish (es), German (de), French (fr) and Czech (cs)), with Dutch (nl) substituted for de in the `st` experiments.

To build phrase-based translation systems, we used the standard Moses (Koehn et al., 2007) training pipeline, in particular employing the usual 5 phrase features – forward and backward phrase probabilities, forward and backward lexical scores and a phrase penalty. The 5-gram Kneser-Ney smoothed language models were trained by SRILM (Stolcke, 2002), with KenLM (Heafield, 2011) used at runtime. The language model is always a linear interpolation of models estimated on the in- and out-of-domain corpora, with weights tuned by SRILM's perplexity minimisation[3]. All experiments were run three times with BLEU scores averaged, as recommended by Clark et al. (2011). Performance was evaluated using case-insensitive BLEU (Papineni et al., 2002), as implemented in Moses.

The baseline systems were tuned using the Moses version of PRO, a reimplementation of the original

[2] `www.opensubtitles.org`

[3] Our method could also be applied to language model interpolation but we chose to focus on phrase tables in this paper.

algorithm using the sampling scheme recommended by Hopkins and May. We ran 15 iterations of PRO, choosing the weights that maximised BLEU on the tuning set. For the PRO training of the interpolated models, we used the same sampling scheme, with optimisation of the model weights and interpolation weights implemented in Python using scipy[4]. The implementation is available in Moses, in the `contrib/promix` directory.

The phrase table interpolation and perplexity-based minimisation of interpolation weights used the code accompanying Sennrich (2012), also available in Moses.

3.2 Results

For each of the two test sets (`nc` and `st`), we compare four different translation systems (three baseline systems, and our new interpolation method):

in Phrase and reordering tables were built from just the in-domain data.

joint Phrase and reordering tables were built from the in- and out-of-domain data, concatenated.

perp Separate phrase tables built on in- and out-of-domain data, interpolated using perplexity minimisation. The reordering table is as for **joint**.

pro-mix As **perp**, but interpolation weights optimised using our modified PRO algorithm.

So the two interpolated models (**perp** and **pro-mix**) are the same as **joint** except that their 4 non-constant phrase features are interpolated across the two separate phrase tables. Note that the language models are the same across all four systems.

The results of this comparison over the 8 language pairs are shown in Figure 1, and summarised in Table 1, which shows the mean BLEU change relative to the **in** system. It can be seen that the **pro-mix** method presented here is out-performing the perplexity optimisation on the `nc` data set, and performing similarly on the `st` data set.

	joint	**perp**	**pro-mix**
`nc`	+0.18	+0.44	+0.91
`st`	-0.04	+0.55	+0.48

Table 1: Mean BLEU relative to **in** system for each data set. System names as in Figure 1

[4] `www.scipy.org`

4 Discussion and Conclusions

The results show that the **pro-mix** method is a viable way of tuning systems built with interpolated phrase tables, and performs better than the current perplexity minimisation method on one of two data sets used in experiments. On the other data set (`st`), the out-of-domain data makes much less difference to the system performance in general, most probably because the difference between the in and out-of-domain data sets in much larger (Haddow and Koehn, 2012). Whilst the differences between **pro-mix** and perplexity minimisation are not large on the `nc` test set (about +0.5 BLEU) the results have been demonstrated to apply across many language pairs.

The advantage of the **pro-mix** method over other approaches is that it directly optimises the measure that we are interested in, rather than optimising an intermediate measure and hoping that translation performance improves. In this work we optimise for BLEU, but the same method could easily be used to optimise for any sentence-level translation metric.

Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement 288769 (ACCEPT).

References

Pratyush Banerjee, Sudip K. Naskar, Johann Roturier, Andy Way, and Josef van Genabith. 2011. Domain Adaptation in Statistical Machine Translation of User-Forum Data using Component Level Mixture Modelling. In *Proceedings of MT Summit*.

Nicola Bertoldi and Marcello Federico. 2009. Domain Adaptation for Statistical Machine Translation from Monolingual Resources. In *Proceedings of WMT*.

R. H. Byrd, P. Lu, and J. Nocedal. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing*, 16(5):1190–1208.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic, June. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012.

Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June. Association for Computational Linguistics.

Colin Cherry and George Foster. 2012. Batch Tuning Strategies for Statistical Machine Translation. In *Proceedings of NAACL*.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online Large-Margin Training of Syntactic and Structural Translation Features. In *Proceedings of EMNLP*.

Jonathan Clark, Chris Dyer, Alon Lavie, and Noah Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of ACL*.

Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2008. IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models. In *Proceedings of Interspeech*, Brisbane, Australie.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, Prague, Czech Republic, June. Association for Computational Linguistics.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative Instance Weighting for Domain Adaptation in Statistical Machine Translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 451–459, Cambridge, MA, October. Association for Computational Linguistics.

Barry Haddow and Philipp Koehn. 2012. Analysing the Effect of Out-of-Domain Data on SMT Systems. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 422–432, Montréal, Canada, June. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, July. Association for Computational Linguistics.

Mark Hopkins and Jonathan May. 2011. Tuning as Ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Philipp Koehn and Josh Schroeder. 2007. Experiments in Domain Adaptation for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 224–227, Prague, Czech Republic, June. Association for Computational Linguistics.

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 *Proceedings of the ACL Demo Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Franz J. Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of ACL*.

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*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Rico Sennrich. 2012. Perplexity Minimization for Translation Model Domain Adaptation in Statistical Machine Translation. In *Proceedings of EACL*.

Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proc. Intl. Conf. on Spoken Language Processing, vol. 2*, pages 901–904.

Jörg Tiedemann. 2009. News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing (vol V)*, pages 237–248. John Benjamins, Amsterdam/Philadelphia.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online Large-Margin Training for Statistical Machine Translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 764–773, Prague, Czech Republic, June. Association for Computational Linguistics.

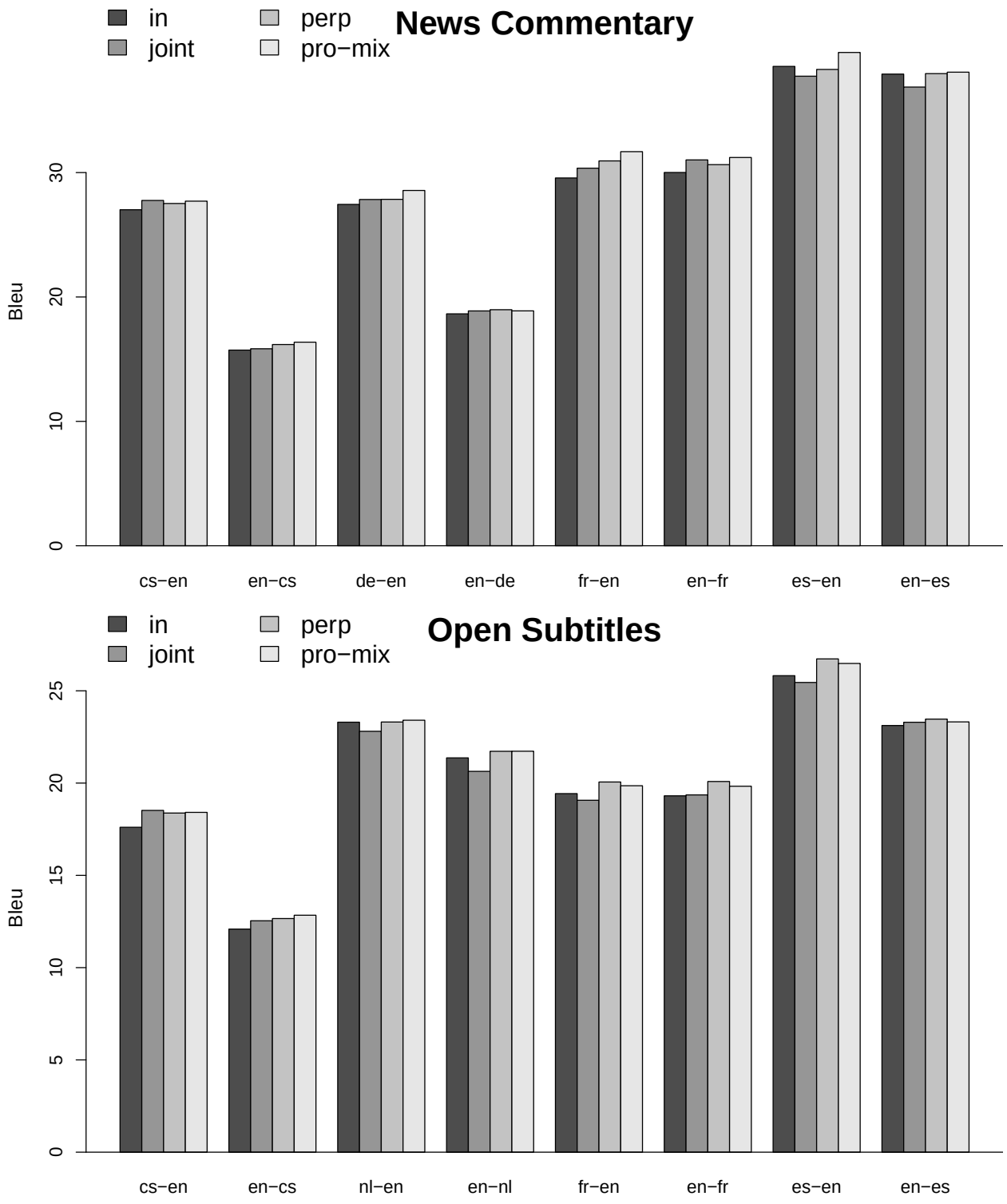

Figure 1: Comparison of the performance (BLEU) on in-domain data, of our **pro-mix** interpolation weight tuning method with three baselines: **in** using just in-domain parallel training data training; **joint** also using europarl data; and **perp** using perplexity minimisation to interpolate in-domain and europarl data.