One System, Many Domains:
Open-Domain Statistical Machine Translation via Feature Augmentation

Jonathan H. Clark **Alon Lavie** **Chris Dyer**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{jhclark,alavie,cdyer}@cs.cmu.edu

Abstract

In this paper, we introduce a simple technique for incorporating domain information into a statistical machine translation system that significantly improves translation quality when test data comes from multiple domains. Our approach augments (conjoins) standard translation model and language model features with domain indicator features and requires only minimal modifications to the optimization and decoding procedures. We evaluate our method on two language pairs with varying numbers of domains, and observe significant improvements of up to 1.0 BLEU.

1 Introduction

Machine translation systems are often used for information assimilation, which allows users to make sense of information written in various languages they do not speak. This use case is particularly important for translation web services, such as Google Translate and Microsoft Bing Translator, which seek to make more of the web accessible to more users. A crucial challenge facing such systems is that they must translate documents from a variety of different domains, but it has been observed that the performance of statistical systems can suffer substantially when testing conditions deviate from training conditions (Bertoldi and Federico, 2009).

However, it is not always possible or cost-effective to collect training data for all of the desired application domains. In fact, training data tends to be collected opportunistically from any available sources rather than from curated sources that match the distribution of test data (Koehn and Schroeder, 2007). As a result, inputs from each application domain may frequently be very different from the training data. With such domain mismatches being commonplace, this paper looks at a way of adapting the behavior of a translation system based on the *domain* of the input documents, which can be matched during both tuning and test time.

One of the key trade offs in designing a statistical model is the balance between bias and variance. In domain adaptation, we would like to *bias* each domain's model toward the distribution of that specific domain, yet we also desire models with low *variance*. Since each domain has less tuning data individually versus the aggregation of all the domains, this gives us statistically less reliable estimates for each domain's parameters, potentially resulting in higher variance.

So what options do we have for making this trade off when faced with D domains? If we desire lower variance, we may ignore the domains entirely and build a single system with F features; however, it will suffer from a bias toward some average over the domains. Another design would be to separately optimize a set of feature weights for each of the D domains, effectively creating D independent translation systems. This strategy expects that each component feature (the language model, lexical probabilities, etc.) would have varying contributions among the different domains. However, from an optimization perspective, this means that we would have D distinct sets of domain-specific tuning data and that the optimization procedures for each domain cannot share statistics among the domains. In

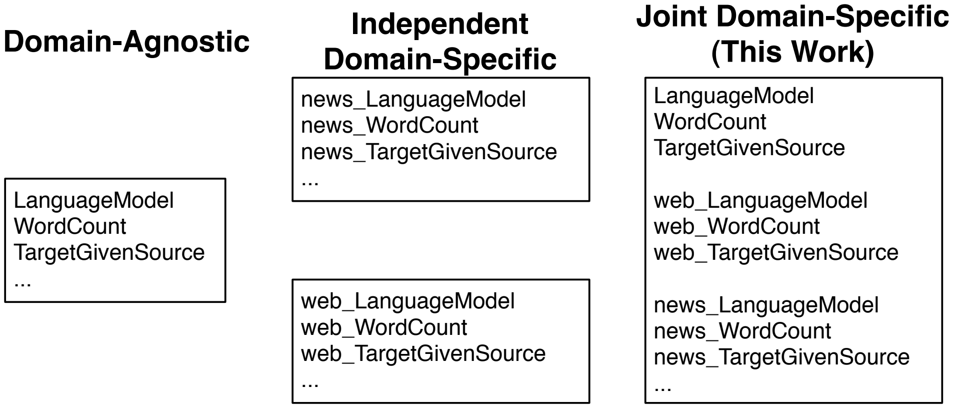


Figure 1: A comparison of feature sets from a baseline domain-agnostic system (left), the features sets of a approach to domain adaptation using D domain-specific systems (middle), and the feature set used in this work (right). In all cases, each feature type (e.g. `LanguageModel`) may have many instances (e.g. `news_LanguageModel`, `web_LanguageModel`), but all instances of the feature have the same *value* – however, in the right two scenarios, each feature instance may have a different *weight*. Only 3 of the 7 typical baseline features (Section 3) are shown for brevity.

this scenario, we would suffer from higher variance – each model's parameters would be more sparsely estimated while removing the possiblity of relying on the more reliably estmated features of the non-adapted model.

In this work, we propose a solution that allows us to incrementally make this trade off in a soft, principled way, without entirely committing to one extreme or the other. First, we describe a simple method of feature augmentation in which we create a single system with $F(D+1)$ features and optimize its weights using a regularization strategy that allows the optimizer to prefer the better-estimated domain-agnostic features (§2). Next, we evaluate our approach on a Czech→English system and a Arabic→English system (§3) and observe a significant improvement of up to 1.0 BLEU, controlling for both test set variation and optimizer instability (§4). Finally, we compare our work with previous approaches (§5) and conclude (§6).

2 Approach

Our approach can be effectively described as two modifications to a conventional statistical system:

1. For each sentence in the tuning set and evaluation set, annotate it with its domain as an observable indicator feature such as `news=true`
2. In addition to the original features, conjoin every feature in the initial model (e.g. `WordCount=3`, `LangaugeModel=0.9`) with the domain indicator feature, resulting in features such as `news_WordCount=3` and `news_LanguageModel=0.9` (Figure 1)

Unlike approaches that build D domain-specific systems and optimize weights for each of them separately (the middle scenario in Figure 1), our approach allows the optimizer to be informed by the data from all domains when estimating weights for the domain-agnostic features shared by all domains (the features at the top right of Figure 1). This framework also opens up new possibilities including having multiple granularities of domains such as `news` and `news-political` retaining the coarser features to combat data sparsity while using the finer features to fit the data more tightly. Similarly, we could encode multiple communication mediums in combination with topics as in `news-political` and `blog-political`.

Similarly, Daumé III (2007) describes a simple kernelized method for feature augmentation in which the features of each data point are labelled as either "general domain" or with a specific target domain. Like Daumé, we accomplish domain adaptation via feature augmentation. However, SMT mod-

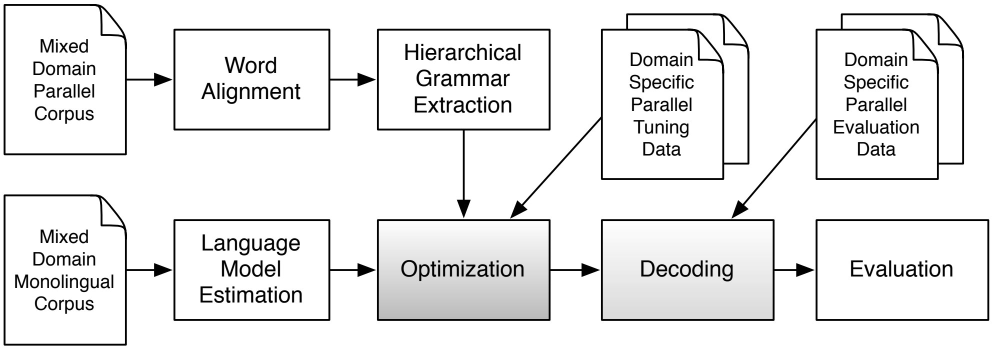

Mixed Domain Parallel Corpus
Word Alignment
Hierarchical Grammar Extraction
Domain Specific Parallel Tuning Data
Domain Specific Parallel Evaluation Data
Mixed Domain Monolingual Corpus
Language Model Estimation
Optimization
Decoding
Evaluation

Figure 2: Our experimental pipeline. Shaded components are affected by this work with the optimization component being the primary focus. The optimizer and its dependent steps are run multiple times to control for optimizer instability (Clark et al., 2010).

els are more complex than those studied by Daumé. Particular characteristics that could effect performance include:

- separate training data sets for the (closed-form) estimation the huge number of parameters in the translation model and language model

- a much smaller tuning data set for estimating the relatively few number of parameters in the final linear model that aggregates the features exposed by these component models[1]

- non-local features (e.g. the language model), which require special handling beyond the simple preprocessing step proposed by Daumé

Our approach is distinguished from previous approaches primarily in its simplicity. We do not construct domain-specific translation models nor domain-specific language models – our feature augmentation affects only the tuning and evaluation data, not the training data (Figure 2). Instead, we merely estimate domain-specific weights for each feature (in addition to domain-agnostic weights). However, we do this estimation jointly for all domains such that the general domain weights can be estimated using the tuning data from all domains.

2.1 Ease of Implementation

Translation Model Augmentation: For a batch system in which we have access to all of the sentences to be translated ahead of time, we can first extract sentence-level grammars.[2] In this case, feature augmentation of the translation model can be implemented as a preprocessing step similar to the "10 lines of Perl" described by Daumé (2007): Each sentence's grammar will have additional features appended to each grammar rule such that they contain a general domain version of each feature and a feature conjoined with the current sentence's domain. For realtime systems,[3] in which we do not have sentence-level translation models, the feature augmentation must be performed at runtime, requiring a simple decoder modification.

Language Model Augmentation: Given that sentence-level language models are generally not used,[4] domain augmentation of the language model feature cannot be implemented as a preprocessing step. Therefore, the domain of each input sentence must be passed to the decoder, and the language model must be modified such that it returns two features for each sentence: `LanguageModel` $= p$ and `domain-LanguageModel` $= p$ where p is the lan-

[1] For example, the translation model is one component model – its parameters are estimated from the parallel training data, and it typically exposes about 7 features to the aggregate model

[2] This is the default usage pattern for the cdec decoder

[3] For example, Moses does not use sentence-level phrase tables.

[4] The authors are not aware of any major decoders that support sentence-level language models. This is likely due to sentence-level LMs being impractical due to their large size, which results from the large space of target translations.

guage model probability of each partial hypothesis. In the case of the cdec decoder, which was used to implement this work, the modification involved only a few lines of code.

2.2 Pairwise Ranking Optimization

As the number of domains increases, feature augmentation can result in much larger feature sets to be optimized. While MERT has proven to be a strong baseline, it does not scale to larger feature sets in terms of both inefficiency and overfitting. While MIRA (Chiang et al., 2008) has been shown to be effective over larger feature sets, as an on-line margin learning algorithm, it is more difficult to regularize – this will become important in Section 2.4. Therefore, we use the PRO optimizer (Hopkins and May, 2011) as our baseline learner. PRO works by sampling pairs of hypotheses from the decoder's k-best list and then providing these pairs as a binary training examples to a standard binary classifier to obtain a new set of weights for the linear model. In our case, the binary classifier is trained using L-BFGS. This procedure of sampling training pairs and then optimizing the pairwise rankings is repeated for a specified number of iterations. It has been shown to perform comparably to MERT for a small number of features, and to significantly outperform MERT for a large number of features (Hopkins and May, 2011; Ganitkevitch et al., 2012).

2.3 Impact on Time and Space Requirements

In this section, we describe the minimal impact that this technique has on development and runtime time and space requirements. During system development, no additional time nor space is required for building additional translation models or language models since we construct only one per language pair. This also holds at runtime, which can be important if multiple deployed translation systems are competing for CPU and RAM resources on a shared server.

The main burden introduced by feature augmentation is the larger number of features itself. As discussed above in Section 2.2, this is not an issue for PRO since it scales robustly to very large feature sets. However, MERT would likely fare poorly as its time complexity scales linearly with the number of features in the model.

The space requirement induced by these extra features is minimal. As noted by Daumé (2007), for an initial feature set with F features our feature space is in $\mathbb{R}^F$. Then for 2 domains, this gives us a feature space $\mathbb{R}^{3F}$ or for D domains, our feature vector will be in $\mathbb{R}^{(D+1)F}$ where the constant of 1 represents the background domain while the $1 \ldots D$ sets of additional parameters correspond to specific domains.

2.4 Feature Complexity Regularization

We note that in the absence of other evidence we prefer to give weight to the background features, since they have more data to estimate them and are therefore more likely to generalize well to unseen data. We encode this preference as a regularizer (equivalently, as a prior). Let $\mathbf{w}$ be the weight vector and let the function $\texttt{SPECIFIC}(h)$ return true iff a feature h is a domain-specific feature. Then we define the regularization hyperparameter γ and regularization term in our objective function as:

$$\mathcal{R}_{\text{complexity}} = \gamma \sum_{\substack{i=0 \\ \text{iff } \texttt{SPECIFIC}(h_i)}}^{|\mathbf{h}|} ||w_i||_2 \qquad (1)$$

Since γ is constant with regard to $\mathbf{w}$, this regularization term differs from the ℓ_2 norm by only a constant. Alternatively, we can view complexity regularization as partitioning the feature set into 2 groups (domain-agnostic and domain-specific) and then applying a ℓ_2 regularizer with weight γ to the domain-specific group. This preserves the convexity of the objective function and makes it easy to incorporate into the gradient-based updates of PRO. With this in mind, the gradient is:

$$\nabla_{\mathbf{w}} \mathcal{R}_{\text{complexity}} = \gamma \sum_{\substack{i=0 \\ \text{iff } \texttt{SPECIFIC}(h_i)}}^{|\mathbf{h}|} 2w_i \qquad (2)$$

We apply this penalty to the domain-specific features *in addition* to the default ℓ_2 regularizer.

3 Experimental Setup

Formalism: In our experiments, we use a hierarchical phrase-based translation model (Chiang, 2007). A corpus of parallel sentences is first word-aligned, and then phrase translations are extracted heuristically. In addition, hierarchical grammar rules are extracted where phrases are nested. Such aligned subphrases are used to generalize their parent phrases by being substituted as a single non-terminal symbol [X]. In general, our choice of formalism is rather unimportant – our techniques should apply to most common phrase-based and chart-based paradigms including Hiero and syntactic systems. Our decision to use Hiero was primarily motivated by the cdec decoder's API being most amenable to implenting these techniques.

Decoder: For decoding, we will use cdec (Dyer et al., 2010), a multi-pass decoder that supports syntactic translation models and sparse features.

Optimizer: Optimization is performed using PRO (Hopkins and May, 2011) as implemented by the cdec decoder. We run PRO for 30 iterations as suggested by Hopkins and May (2011), though analysis indicates that the parameters converged much earlier. The PRO optimizer internally uses a L-BFGS optimizer with the default ℓ_2 regularization implemented in cdec. Any additional regularization (as described in Section 2.4) is explicitly noted.

Baseline Features: We use the baseline features produced by Lopez' suffix array grammar extractor (Lopez, 2008a; Lopez, 2007; Lopez, 2008b), which is distributed with cdec:

- $\log P_{\text{coherent}}(e|f)$: The coherent phrase-to-phrase translation probability (Lopez, 2008b, p. 103). The phrasal probability of each English SCFG antecedent (e.g. "el [X] gato") given a particular foreign SCFG antecedent "the [X] cat" combined with *coherence*, the ratio of successful source extractions to the number of attempted extractions
- $\log P_{\text{lex}}(e|f)$, $\log P_{\text{lex}}(f|e)$: The lexical alignment probabilities within each translation rule, as computed by a maximum likelihood estimation over the Viterbi alignments

	Sentences	Translations
NIST Train		
Total	5.4M	1
MT06 (tune)		
Newswire	1033	4
Web	764	4
Total	1797	4
MT08 (test)		
Newswire	813	4
Web	547	4
Total	1360	4
MT09 (test)		
Newswire	586	4
Web	727	4
Total	1313	4

Figure 3: Corpus statistics for Arabic→English experiments. Note that the training data was sampled in different quantities to create learning curves, simulating lower data scenarios.

- $\log P_{\text{LM}}(e)$: The log probability of the target translation hypothesis under a language model
- $c(e)$ The count of target words (terminals) in the target translation hypothesis
- $c(\texttt{glue})$ The count of glue rules used in the derivation
- $c(\texttt{OOV}_{\text{TM}})$ The count of source tokens that were not recognized by the translation model (out of vocabulary) and were therefore passed through
- $c(\texttt{OOV}_{\text{LM}})$ The count of target tokens that were not recognized by the language model (out of vocabulary)

Domain-Specific Features: In addition to the baseline features, as standalone features with each of the 7 features conjoined with each of the D domains, for a total of $7(D+1)$ features overall.

Complexity Regularization: In our experiments, we used a hyperparameter value of $\gamma = 5000$, 10 times the default global ℓ_2 regularizer of 500. This value worked well and so no tuning (manual or otherwise) was performed.

		Sentences
CzEng *(Sections 1-97)**	Train	1M

	Tune (Sents) *Sec 98*	Test (Sents) *Sec 99*
Fiction	500	1000
EU Legislation	500	1000
Subtitles	500	1000
Parallel Web	500	1000
Tech Docs	500	1000
News	500	1000
Total	3000	6000

Figure 4: Corpus statistics for CzEng→English experiments. The tuning and test sets have one reference. We use the * to indicate that sections were sampled to create a more balanced data set. Project Navajo data was omitted due to its small size. No domain-specific information from the training set was used in the construction of our models.

Arabic Resources: We build an Arabic→English system, training on the large NIST MT 2009 constrained training corpus[5] of approximately 5 million sentence pairs with about 181 million English words. We tune on the NIST MT 2006 dataset and test on NIST MT 2008 and 2009.[6] Full details are shown in Figure 3.

Czech resources: We also construct a Czech→English system based on the CzEng 1.0 data (Bojar et al., 2012). Both sides of the data were lowercased as a preprocessing step. For our experiments, we sampled a training set of 1M sentences and training and tuning sets that are evenly balanced among the various domains contained in CzEng. Full details are shown in Figure 4.

	Baseline	Domain Augmented
	MT08	
News	47.8	47.8 (±)
Web	30.5	**31.0** (+0.5)
All	40.5	**40.7** (+0.2)
	MT09	
News	51.6	51.5 (±)
Web	31.6	**32.3** (+0.7)
All	41.9	**42.2** (+0.3)

Figure 5: Results of multi-domain experiments on the large NIST MT Arabic→English data set as measured by the BLEU metric. The domain augmented system has 21 features. Both the Meteor and TER evaluation metrics agreed with BLEU in the conditions where BLEU improved. Boldfaced results indicate significance at the 0.01 level according to approximate randomization over 5 optimizer replications.

	Baseline	Domain Augmented
Test (All Domains)	46.5	**47.5** (+1.0)

Figure 6: Results of feature augmentation experiments on the 1M sentence Czech→English data set as measured by the BLEU metric. The domain augmented system has 49 features. Both the Meteor and TER evaluation metrics also showed improvements. Results are significant at the 0.01 level according to approximate randomization over 3 optimizer replications.

Evaluation: We quantify increases in translation quality using automatic metrics including BLEU (Papineni et al., 2002). We control for test set variation and optimizer instability by measuring statistical significance according to approximate randomization (Clark et al., 2010).[7] Evaluation is performed on tokenized lowercased references.

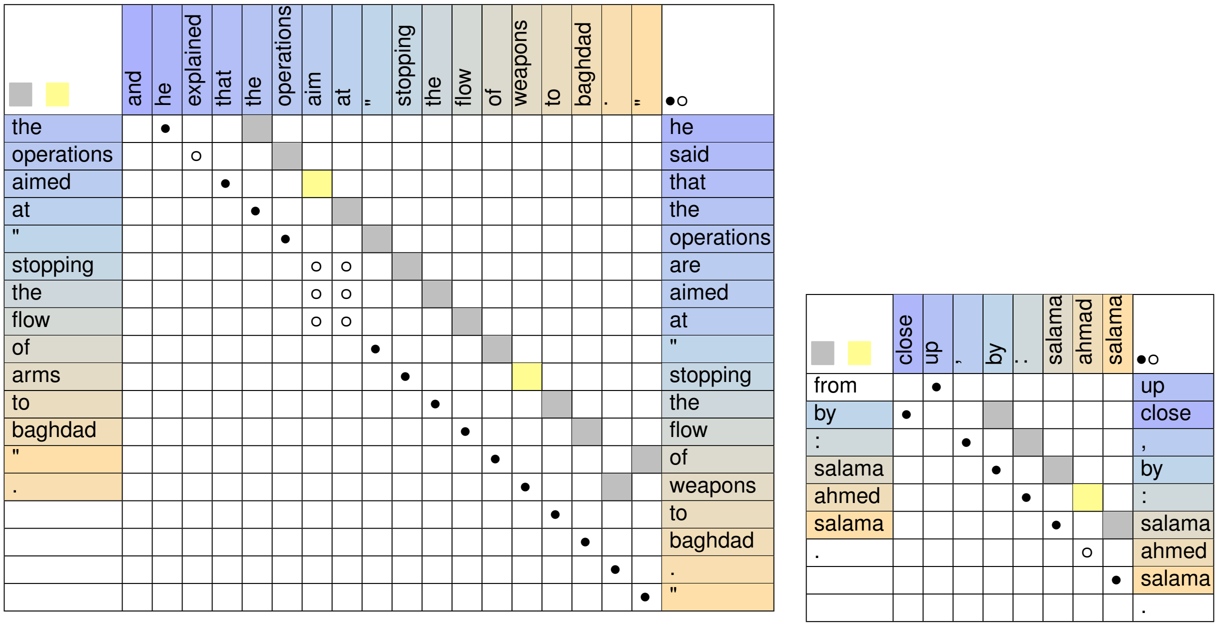
4 Results and Analysis

We show the results of using feature augmentation for domain adaptation to an Arabic→English system in Figure 5. There, we see an overall improvement of only 0.2 - 0.3 BLEU. Interestingly, we see an improvement of up to 0.7 BLEU in the weblog domain while we see no improvement in the newswire do-

[5] A list of the resources available as part of the NIST MT 2009 constrained training resources is available at `http://www.itl.nist.gov/iad/mig/tests/mt/2009/MT09_ConstrainedResources.pdf`

[6] The NIST MT test sets are available from the LDC as catalog numbers LDC2010T{10,11,12,13,17,21,23}. One of the four references for the Arabic MT08 weblog data was not processed correctly in the officially released XML document and is mismatched with regard to the source sentences. There is no obvious way of reversing this error. However, since three references are still valid, this should have negligible impact on the results.

[7] MultEval 0.4.2 is available at `github.com/jhclark/multeval`



Segment 207

P:	0.931	vs	0.920	:	-0.011
R:	0.756	vs	0.881	:	0.125
Frag:	0.441	vs	0.338	:	-0.102
Score:	0.435	vs	0.587	:	0.152

Segment 341

P:	0.754	vs	0.871	:	0.117
R:	0.613	vs	0.925	:	0.312
Frag:	0.435	vs	0.493	:	0.058
Score:	0.356	vs	0.465	:	0.108

Figure 7: A Meteor X-Ray visualization of 2 improved examples, cherry picked from the MT08 newswire test set. The alignment to the reference (top border) for the baseline system (left border) is shown as shaded blocks while the alignment between the reference and the augmented system (right border) is shown using dots. Improvements are generally subtle, but consistent.

main. We suspect this is due to the domain of the training data being more aligned with the newswire data.

However, on the Czech→English system in Figure 6, we see an improvement of 1.0 BLEU on the test set consisting of the 6 domains listed in Figure 4. In this case, we suspect that the much larger number of domains hinders the domain agnostic system from being able to tightly fit to any of the individual domains as well as the domain augmented system.

To gain a deeper understanding of the changes in translation quality resulting from our modifications, we used the Meteor X-Ray visualization tool (Denkowski and Lavie, 2010). We present two improved examples from the MT08 newswire test set in Figure 7. In general, improvements follow the pattern of these examples, remaining subtle yet consistent.

5 Related Work

Domain adaptation in statistical machine translation has been widely investigated.

Component model adaptation: Perhaps the largest body of work focuses on adapting the translation model and language model. Koehn and Schroeder (2007) explore several techniques for domain adaptation in SMT including multiple translation models (via multiple factored decoding paths), interpolated langauge models, and multiple language models. Xu et al. (2007) build a general domain translation system and then construct domain-specific language models and tune domain-specific feature weights to aggregate the component models. Unlike the work in this paper, the optimization procedure for training these final feature weights cannot share statistics among domains. Foster and Kuhn (2007) train independent models on each domain and use a mixture model (both linear and log-linear) to weight the component models (both the translation model and language model) appropriately for the current context. This was later extended by Foster et al. (2010) to examine fine-grained instance-level characteristics rather than requiring each domain to have a distinct model.

Word alignment: Civera and Juan (2007) explore an extension of the HMM alignment model that performs domain adaptation using mixture modelling.

Automatic Post-Editing: Isabelle et al. (2007) use an automatic post-editor to accomplish domain adaptation, effectively "translating" from a domain-agnostic version of the target language into a domain-adapted version of the target language.

Monolingual data: Others have used monolingual data to improve in-domain performance. Ueffing et al. (2007) use source monolingual data in various ways to improve target domain performance, focusing on corpus filtering techniques such that more relevant data is included in the models. Bertoldi and Federico (2009) saw large improvements by using automatic translations of monolingual in-domain data to augment the training data of their original system.

Domain identification: Closely related to domain adaptation is domain identification. Banerjee et al. (2010) focus on the problem of determining the domain of the input data so that the appropriate domain-specific translation system can be used.

EBMT: Phillips (2012) describes the Cunei framework, which natively uses instance-based features to determine which translation examples are most relevant given the current context. One major application area of this framework is domain adaptation.

Multi-Task Learning: Simianer et al. (2011) propose a variant of minimum error rate training (MERT) that allows for multi-task learning. Like our work, multi-task MERT allows sharing of common parameters among various task-specific optimization problems.

NLP: Other areas of NLP have also seen work on domain adaptation. For instance, Dredze et al. (2007) report their experiments in a shared task for domain adaptation of dependency parsing in which they explored adding features more likely to transfer across domains and removing features less likely to transfer.

Machine learning: Domain adaptation has also been well-studied from a more general machine learning perspective. Daumé (2006) points out that "the most basic assumption used in statistical learning theory is that training data and test data are drawn from the same underlying distribution" and goes on to formuvvlate the MEGA (Maximum Entropy Genre Adaptation) model. Blitzer et al. (2006; Blitzer (2007) describe structural correspondence learning, which automatically discovers features that behave similarly in both a source and target domain. Ben-David et al. (2010) provide formal bounds on source and target error for various discriminative learning criteria. Huang and Yates (2012) propose a domain adaptation method of representation learning, which automatically discovers feature more likely to generalize across domains. By using posterior regularization to bias the process of representation learning, they observe improvements on a part of speech tagging task and a named entity recognition task. Domain adaptation has also been framed as a semi-supervised learning problem in which unlabelled in-domain data is used to augment out-of-domain labelled data (Daumé III et al., 2010).

Perhaps most similar to this work is Daumé (2007), which proposes a method for feature augmentation in which the features of each data point augmented with a label of either "general domain" or a specific target domain.

6 Conclusion

In this paper, we presented a very simple technique for performing domain adaptation using feature augmentation, which resulted in improvements of over 1.0 BLEU points on multiple language pairs and multiple data sets. In the future, we hope to apply this technique to domains of varying granularity and combine it with more complex feature sets.

Acknowledgments

The authors wish to thank Ondrej Bojar and Greg Hanneman for their help. This research was supported in part by the National Science Foundation through XSEDE resources provided by the San Diego Supercomputer Center. This work was supported by a Google Faculty Research Award.

References

Pratyush Banerjee, Jinhua Du, Baoli Li, Sudip Kr Naskar, Andy Way, and Josef Van Genabith. 2010. Combining Multi-Domain Statistical Machine Translation Models using Automatic Classifiers. In *Association for Machine Translation in the Americas*.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A Theory of Learning from Different Domains. *Machine Learning*, 79(1-2):151–175, October.

Nicola Bertoldi and Marcello Federico. 2009. Domain Adaptation for Statistical Machine Translation with Monolingual Resources. In *Workshop on Statistical Machine Translation*, number March, pages 182–189.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain Adaptation with Structural Correspondence Learning. In *Empirical Methods in Natural Language Processing*, number July, pages 120–128, Sydney.

John Blitzer. 2007. *Domain Adaptation of Natural Language Processing Systems*. Ph.D. thesis, University of Pennsylvania.

Ondej Bojar, Zdeněk Žabokrtský, Ondej Dušek, Petra Galuščáková, Martin Majliš, David Mareček, Jií Maršík, Michal Novák, Martin Popel, and Aleš Tamchyna. 2012. The Joy of Parallelism with CzEng 1 .0. In *Proceedings of LREC2012*, Istanbul, Turkey. European Language Resources Association.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 224–233, Morristown, NJ, USA. Association for Computational Linguistics.

David Chiang. 2007. Hierarchical Phrase-Based Translation. *Computational Linguistics*, 33(2):201–228, June.

Jorge Civera and Alfons Juan. 2007. Domain Adaptation in Statistical Machine Translation with Mixture Modelling. In *Workshop on Statistical Machine Translation*, number June, pages 177–180, Prague.

Jonathan H Clark, Chris Dyer, Alon Lavie, and Noah A Smith. 2010. Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability. In *Association for Computational Linguistics*.

Hal Daumé III and Daniel Marcu. 2006. Domain Adaptation for Statistical Classifiers. *Journal of Artificial Intelligence Research*, 26:101–126.

Hal Daumé III, Abhishek Kumar, and Avishek Saha. 2010. Co-regularization Based Semi-supervised Domain Adaptation. In *Advances in Neural Information Processing*, pages 1–9.

Hal Daumé III. 2007. Frustratingly Easy Domain Adaptation. In *Association for Computational Linguistics*, number June, pages 256–263, Prague.

Michael Denkowski and Alon Lavie. 2010. Extending the METEOR Machine Translation Evaluation Metric to the Phrase Level. In *North American Association for Computational Linguistics*.

Mark Dredze, John Blitzer, Partha Pratim Talukdar, Kuzman Ganchev, V João Graça, and Fernanda Pereira. 2007. Frustratingly Hard Domain Adaptation for Dependency Parsing. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*.

Chris Dyer, Jonathan Weese, Adam Lopez, Vladimir Eidelman, Phil Blunsom, and Philip Resnik. 2010. cdec: A Decoder, Alignment, and Learning Framework for Finite-State and Context-Free Translation Models. In *Association for Computational Linguistics*, number July, pages 7–12.

George Foster and Roland Kuhn. 2007. Mixture-Model Adaptation for SMT. In *Proceedings of the Workshop on Statistical Machine Translation*.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative Instance Weighting for Domain Adaptation in Statistical Machine Translation. In *Empirical Methods in Natural Language Processing*, number October, pages 451–459.

Juri Ganitkevitch, Yuan Cao, Jonathan Weese, Matt Post, and Chris Callison-Burch. 2012. Joshua 4.0: Packing, PRO, and Paraphrases. In *Workshop on Statistical Machine Translation*, pages 283–291.

Mark Hopkins and Jonathan May. 2011. Tuning as Ranking. *Computational Linguistics*, pages 1352–1362.

Fei Huang and Alexander Yates. 2012. Biased Representation Learning for Domain Adaptation. In *Empirical Methods in Natural Language Processing*.

Pierre Isabelle, Cyril Goutte, and Michel Simard. 2007. Domain adaptation of MT systems through automatic post-editing. In *Machine Translation Summit XI*, Copenhagen.

Philipp Koehn and Josh Schroeder. 2007. Experiments in Domain Adaptation for Statistical Machine Translation. In *Workshop on Statistical Machine Translation*, number June, pages 224–227, Prague.

Adam Lopez. 2007. Hierarchical Phrase-Based Translation with Suffix Arrays. *Computational Linguistics*, (June):976–985.

Adam Lopez. 2008a. Tera-Scale Translation Models via Pattern Matching. In *Association for Computational Linguistics Computational Linguistics*, number August, pages 505–512.

Adam David Lopez. 2008b. *Machine Translation by Pattern Matching*. Ph.D. thesis, University of Maryland.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Computational Linguistics*, number July, pages 311–318.

Aaron B. Phillips. 2012. *Modeling Relevance in Statistical Machine Translation: Scoring Alignment, Context, and Annotations of Translation Instances*. Ph.D. thesis, Carnegie Mellon University.

Patrick Simianer, Katharina Wäschle, and Stefan Riezler. 2011. Multi-Task Minimum Error Rate Training for SMT. *Prague Bulletin of Mathematical Linguistics*, (96):99–108.

Nicola Ueffing, Haffari Gholamreza, and Anoop Sarkar. 2007. Transductive learning for statistical machine translation. In *Workshop on Statistical Machine Translation2*.

Jia Xu, Yonggang Deng, Yuqing Gao, and Hermann Ney. 2007. Domain Dependent Statistical Machine Translation. In *MT Summit*.