Translation Model Interpolation for Domain Adaptation in TectoMT

Rudolf Rosa, Ondřej Dušek, Michal Novák, Martin Popel
Charles University in Prague, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, Prague, Czech Republic
{rosa,odusek,mnovak,popel}@ufal.mff.cuni.cz

Abstract

We present an implementation of domain adaptation by translation model interpolation in the TectoMT translation system with deep transfer. We evaluate the method on six language pairs with a 1000-sentence in-domain parallel corpus, and obtain improvements of up to 3 BLEU points. The interpolation weights are set uniformly, without employing any tuning.

1 Introduction

Statistical machine translation (SMT) is now a well-established field of natural language processing, with many real-world applications. The core of an SMT system is the translation model (TM), created from parallel data. For many language pairs, especially those where one member of the pair is English, parallel data in several domains are often abundant; typical examples are legal texts (e.g. Europarl), film subtitles, books, and newspapers. Thus, it is usually easy to build SMT systems for these domains with reasonable performance.

For other domains, quite the opposite is often true – the amount of in-domain parallel data is low, which limits the accuracy of translation systems trained on such data. Therefore, the small in-domain data are typically combined with larger available out-of-domain data. The simplest method that can be employed is data concatenation, where all the available parallel data are merged and used to train one TM. However, this method is not optimal (Daumé III, 2009) because the TM is usually biased towards translations that are more frequent in the merged data, which are often translations from the larger out-of-domain data; the effect of the small in-domain data tends to be "washed out".

Several authors (see Section 5) have instead successfully employed the method of TM interpolation, in which in-domain and out-of-domain TMs are created separately, and linear interpolation is then used to obtain the final TM. As each of the TMs can be assigned a different weight, it is possible to promote the in-domain TM, effectively biasing the decoder towards the target domain.

In our work, we successfully implement domain adaptation by TM interpolation in the TectoMT system, a hybrid SMT system based on deep language processing and deep transfer. We apply the system to translation of user requests and helpdesk answers in the information technologies (IT) domain, with only 1000 in-domain parallel sentences available, in addition to large out-of-domain data. For several reasons, we use uniform interpolation weights without any tuning (see Section 3). We show our method to be very successful, with the interpolated model achieving improvements of several BLEU points over the individual TMs across six translation directions: EN↔CS, EN↔ES, EN↔NL (English to and from Czech, Spanish and Dutch).

We briefly present TectoMT in Section 2. In Section 3, we describe our implementation of domain adaptation by model interpolation. Section 4 evaluates our method using the QTLeap IT helpdesk corpus, Section 5 reviews related work, and Section 6 concludes the paper and presents directions for future research.

Proceedings of the 1st Deep Machine Translation Workshop (DMTW 2015), pages 89–96,
Praha, Czech Republic, 3–4 September 2015.

2 TectoMT System

TectoMT is a structural machine translation system with a tree-to-tree transfer on the deep syntax layer, first introduced by Žabokrtský et al. (2008). It is based on the Prague "tectogrammatics" theory of Sgall et al. (1986). The system uses two layers of structural description with dependency trees: surface syntax (*a-layer*, *a-trees*) and deep syntax (*t-layer*, *t-trees*).

The analysis phase is two-step and proceeds from plain text over a-layer to t-layer (see Section 2.1). The transfer phase of the system is based on maximum entropy context-sensitive translation models (Mareček et al., 2010) and Hidden Markov Tree Models (Žabokrtský and Popel, 2009) (see Section 2.2). The subsequent generation phase consists of rule-based components that gradually change the deep target language representation into a shallow one, which is then converted to text (see Section 2.3).

2.1 Analysis

The analysis phase consists of a pipeline of standard NLP tools that perform the analysis to the a-layer, followed by a rule-based conversion to t-layer.

In the analysis pipeline, the input is first segmented into sentences and tokenized using rule-based modules from the Treex toolkit[1] (Popel and Žabokrtský, 2010). A statistical part-of-speech tagger and dependency parser are applied to the tokenized sentences and conclude the a-layer analysis part.[2] The a-trees contain one node for each token of the sentence with its surface word form and the lemma (base form), its part-of-speech/morphology, and its surface dependency label.

A t-tree is a dependency tree where only content words (nouns, full verbs, adjectives, adverbs) and coordinating conjunctions have their own nodes; grammatical words such as prepositions or auxiliary verbs are hidden. Each node has the following attributes:

- *t-lemma* – deep lemma,
- *functor* – a deep-syntactic/semantic role label,
- *formeme* – a concise description of its morpho-syntactic surface form (Dušek et al., 2012), e.g., `v:fin` for a finite verb or `n:in+X` for a noun in prepositional phrase with the preposition *in*,
- *grammatemes* – a set of deep grammatical attributes, covering properties such as tense, gender, number, person, or modality.

T-trees are created from a-trees using a set of rules which collapse auxiliaries and assign all the required attributes to each t-node.

2.2 Transfer

In the transfer phase, an initial target t-tree is obtained as a copy of the source t-tree. Target t-lemmas and formemes of the t-nodes are suggested by a set of TMs, and the other attributes are transferred by a set of rules.

For both t-lemmas and formemes, we use two separate TMs:

- MaxEnt TM – a discriminative model whose prediction is based on features extracted from the source tree. The discriminative TM (Mareček et al., 2010) is in fact an ensemble of maximum entropy (MaxEnt) models (Berger et al., 1996), each trained for one specific source t-lemma/formeme. However, as the number of types observed in the parallel treebank may be too large, infrequent source t-lemmas/formemes are not covered by this type of TM.
- Static TM – this is only a dictionary of possible translations with relative frequencies (no contextual features are taken into account). This model is available for most source t-lemmas/formemes seen in training data.[3]

[1] `http://ufal.mff.cuni.cz/treex` and `https://github.com/ufal/treex`

[2] The modules used for the analysis in the individual languages vary, but all of them follow the same structure. For instance, the English pipeline uses the Morče tagger (Spoustová et al., 2007) and the MST parser (McDonald et al., 2005).

[3] Both the MaxEnt and the Static TM are subject to pruning during training, with a higher threshold used for MaxEnt; see Section 4.2 for more details.

When performing the transfer, the two TMs are combined via interpolation. Each of the models is assigned an interpolation weight – the translation probabilities emitted by the model are multiplied by the model's weight, and weights of both models are normalized to sum up to 1.

After the TMs are applied, each t-tree node contains a list of possible formemes and a list of possible t-lemmas, along with their estimated probabilities. There are two possible ways of combining the lists:

1. Just using the first item of both lists (the simplest way, but its performance may not be ideal since incompatible combinations are sometimes produced).
2. Using a Hidden Markov Tree Model (Žabokrtský and Popel, 2009), where a Viterbi search is used to find the best t-lemma/formeme combinations globally over the whole tree.

In the current TectoMT version, HMTM is only used in EN→CS translation. HMTM for the remaining languages will be added in the near future.

2.3 Synthesis

The synthesis is a pipeline of rule-based modules (Žabokrtský et al., 2008; Dušek et al., 2015) that gradually change the translated t-tree into an a-tree (surface dependency tree), adding auxiliary words and punctuation and resolving morphological attributes. Some basic word-order rules are also applied.

The individual a-tree nodes/words are then inflected using a morphological dictionary (Straková et al., 2014) or a statistical tool trained on an annotated corpus (Dušek and Jurčíček, 2013). The resulting tree is then simply linearized into the output sentence.

3 Domain Adaptation by Model Interpolation

The general approach of domain adaptation by model interpolation is rather simple:

1. Train a TM on out-of-domain data,
2. Train a TM on in-domain data,
3. Interpolate the TMs,
4. Translate using the interpolated TM.

As mentioned in Section 2.2, TectoMT uses four TMs by default – a Static formeme TM, a MaxEnt formeme TM, a Static t-lemma TM, and a MaxEnt t-lemma TM. Therefore, we train this set of four models on each of the datasets.

Even in the original TectoMT pipeline, TM interpolation is used to combine a Static model with a MaxEnt model; however, it only supported interpolation of one Static model with one MaxEnt model. Therefore, we extended the pipeline to allow interpolation of multiple TMs; for each model, one must specify the model file, the type of the model (Static/MaxEnt), and its interpolation weight.

In our setup, we use the default MaxEnt–Static interpolation weights as defined in TectoMT, and we use the same weights for in-domain TMs and out-of-domain TMs. This has a similar effect to training the TMs on concatenated out-of-domain and in-domain data with the in-domain data duplicated as many times as to have the same size as the out-of-domain data (modulo some hard thresholds).

The standard approach, as applied in phrase-based SMT systems, would be to use tuning on an in-domain development set to find a well-performing set of weights, by employing an optimizer such as MERT or PRO. However, we do not apply tuning in our setup our in-domain training dataset is very small (1000 sentences only) and we do not want to further divide it into training and development parts and we had not enough time to apply cross-validation. Still, we believe to be able to perform weight tuning in future, which may lead to additional performance gains.

4 Evaluation

We evaluate our implementation on a task for the QTLeap project. We first describe the datasets used for training and testing our system (in Section 4.1), then list the settings used for training (in Section 4.2), and finally discuss the results we obtained (Section 4.3).

4.1 Dataset

In-domain

Our in-domain data set comes from the QTLeap corpus,[4] which is a set of IT-related user requests ("*q*uestions") and helpdesk responses ("*a*nswers") in English, translated into Basque, Bulgarian, Czech, Dutch, German, Portuguese, and Spanish. In this paper, we only evaluate using Czech, Dutch, and Spanish.

Currently, two 1000-sentence batches are available to us, Batch1 as a development and training set, and Batch2 as a test set (this division is given by the QTLeap project setup). Moreover, the data are not divided into the batches randomly, but sequentially, so they all come from the same domain, but the topics in Batch1 and Batch2 are somewhat different (i.e., the similarity of Batch1 sentences to other Batch1 sentences is greater than the similarity of Batch1 sentences to Batch2 sentences).

For translations into English, we use Batch1q (user requests) as the in-domain training data. For translations from English, we use Batch1a (helpdesk answers) as the in-domain training data. This reflects the inteded purpose of the MT systems and the final application of translating user questions into English and helpdesk answers back to the original language (Czech, Dutch, Spanish).

Out-of-domain

We use the following corpora to train our out-of-domain models (each language contains parallel texts with English):

- Czech – CzEng 1.0 (Bojar et al., 2012), with 15.2 million parallel sentences, containing a variety of domains, including fiction books, news texts, EU legislation, and technical documentation.
- Dutch – A combination of Europarl (Koehn, 2005), Dutch Parallel Corpus (Macken et al., 2007), and KDE technical documentation; 2.2 million parallel sentences in total.
- Spanish – Europarl, containing 2 million parallel sentences.

Monolingual

For Czech as the target language, we used the WMT News Crawl monolingual training data (2007–2012, 26 million sentences in total) to train the HMTM.[5] Other target languages do not use an HMTM (see Sections 2.2 and 4.2).

4.2 Setup

We use the QTLeap TM training makefile[6] to train a Static and a MaxEnt TM on both in-domain and out-of-domain data. As discussed in Section 3, we do not use tuning on development data to set TM pruning thresholds and interpolation weights.

Two thresholds are used to prune the TMs:

- *MinInst* – the minimum number of instances required to train a model for a single source t-lemma/formeme,
- *MinPerClass* – the minimum number of instances for the same target class (translation variant of a t-lemma/formeme) so that this class is included in the classification.

The MaxEnt TM thresholds for the out-of-domain are set higher since much more data (and noise) is available. We used MinInst=100 and MinPerClass=5 for out-of-domain TMs and MinInst=2 and MinPerClass=1 for in-domain TM. The Static TM thresholds are MinInst=2 and MinPerClass=1.

For TM interpolation, we use an identical set of weights for the out-of-domain TM and for the in-domain TM; these are listed in Table 1.

TM for	TM type Static	Maxent
Formemes	1.0	0.5
T-lemmas	0.5	1.0

Table 1: Weights of TMs in interpolation; the same set used both for out-of-domain TMs and in-domain TMs in all translation directions.

Translation	Out-of-domain	In-domain	Interpolation	Improvement
EN→CS	30.60	28.41	**31.27**	+0.67
CS→EN	27.11	21.51	**28.25**	+1.14
EN→ES	20.35	23.28	**26.48**	+3.20
ES→EN	18.50	18.54	**20.44**	+1.90
EN→NL	23.03	21.37	**24.29**	+1.26
NL→EN	37.03	33.68	**38.93**	+1.90

Table 2: Automatic evaluation in terms of BLEU on QTLeap corpus Batch2. Results obtained using out-of-domain TMs only, in-domain TMs only, and the interpolation of both in-domain and out-of-domain TMs. Improvement in BLEU is relative to the better of the Out-of-domain and In-domain results.

4.3 Results and Discussion

The results of our experiments on QTLeap corpus Batch2 are summarized in Table 2 (Batch2q for translations into English, Batch2a for translations from English). They show that for all translation directions, using the interpolation of out-of-domain TMs with in-domain TMs performs better than using any of the two TM types individually. The improvements range from 0.67 BLEU for EN→CS to 3.20 BLEU for EN→ES. We do not have a conclusive explanation for the variation in the amount of the improvement achieved.

In most cases, using (only) the in-domain TM leads to worse results than using (only) the out-of-domain TM. This is to be expected, as the in-domain data are extremely small. Interestingly, for EN→ES, the in-domain TM beats the large out-of-domain TM by nearly 3 BLEU points; in the other direction, the results of the two setups are comparable. We are unsure about the reason behind that.

5 Related Work

A seminal work on domain adaptation by Daumé III (2009) lists eight approaches:

- SRCONLY, TRGONLY, LININT – these correspond to our experiments (using out-of-domain model only, in-domain-model only, and a linear interpolation of both, respectively), but the linear interpolation constant is tuned on a development set.
- ALL – concatenation of training data.
- WEIGHT – as ALL, but the out-of-domain training examples are downweighted so the in-domain examples (which are typically much fewer) have bigger effect on the resulting model. The weight is chosen by cross-validation.
- PRED – the prediction of the out-of-domain model is used as an additional feature for training the final model on the in-domain data.
- PRIOR – out-of-domain weights are used as a prior (via the regularization term) when training the final model on the in-domain data (Chelba and Acero, 2004).

[4] `http://metashare.metanet4u.eu/go2/qtleapcorpus`
[5] `http://www.statmt.org/wmt13/translation-task.html`
[6] See `cuni_train/Makefile` in `https://github.com/ufal/qtleap`.

- EASYADAPT (called AUGMENT in the original paper, sometimes referred to as the "Frustratingly Easy Domain Adaptation") – create three variants of each feature: general, in-specific and out-specific; train on concatenation of in- and out-of-domain data, where on in-domain data, the general and in-specific features are active and on the out-of-domain data, the general and out-specific features are active.

Daumé III (2009) showed that EASYADAPT outperforms the other methods (on a variety of NLP tasks, but not including MT) in the cases when TRGONLY outperforms SRCONLY.[7] Otherwise, LININT, PRED and WEIGHT were the most successful methods. In a follow-up work (Daumé III et al., 2010), EASYADAPT was improved to exploit also additional unlabeled in-domain data.

In MT, many different approaches to domain adaptation have been attempted. Similarly to our experiments, authors combine the predictions of two separate (in-domain and general-domain) translation models (Langlais, 2002; Nakov, 2008; Sanchis-Trilles and Casacuberta, 2010; Bisazza et al., 2011) or language models (Koehn and Schroeder, 2007) in phrase-based statistical MT. Others concentrate on acquiring larger in-domain training corpora for statistical MT by selecting data from large general-domain corpora that resemble the properties of in-domain data (e.g., using cross-entropy), thus building a larger *pseudo-in-domain* training corpus. This technique has been used to adapt language models (Eck et al., 2004; Moore and Lewis, 2010) as well as translation models (Hildebrand et al., 2005; Axelrod et al., 2011) or their combination (Mansour et al., 2011; Dušek et al., 2014).

6 Conclusion and Future Work

In this paper, we presented our implementation of machine translation domain adaptation by translation model interpolation in the TectoMT system. We evaluated the method using large out-of-domain parallel data and small in-domain parallel data (1000 sentences) in the domain of computer helpdesk requests and responses, using 6 translation directions. The evaluation showed our method to perform well, achieving improvements up to 3.2 BLEU over using only a single training dataset.

In the coming year, we will obtain additional in-domain data, which will allow us to use a portion of the data for tuning the interpolation weights. We are therefore planning to implement an interpolation weights optimizer for TectoMT and try different domain-adaptation techniques (EASYADAPT, PRED and WEIGHT).

Acknowledgements

This research was supported by the grants GAUK 1572314, GAUK 338915, GAUK 2058214, SVV 260 224, and FP7-ICT-2013-10-610516 (QTLeap). This work has been using language resources developed, stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2010013).

References

A. Axelrod, X. He, and J. Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, United Kingdom. ACL.

A. L. Berger, V. J. Della Pietra, and S. A. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71.

A. Bisazza, N. Ruiz, and M. Federico. 2011. Fill-up versus interpolation methods for phrase-based SMT adaptation. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 136–143, San Francisco, CA, USA. International Speech Communication Association.

O. Bojar, Z. Žabokrtský, O. Dušek, P. Galuščáková, M. Majliš, D. Mareček, J. Maršík, M. Novák, M. Popel, and A. Tamchyna. 2012. The joy of parallelism with CzEng 1.0. In *LREC*, page 3921–3928, Istanbul.

[7] As we can see in Table 2, this is the case of ES→EN (and maybe EN→ES), so we plan to use EASYADAPT there in future.

C. Chelba and A. Acero. 2004. Adaptation of maximum entropy capitalizer: Little data can help a lot. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 285–292, Barcelona, Spain, July. Association for Computational Linguistics.

H. Daumé III, A. Kumar, and A. Saha. 2010. Frustratingly easy semi-supervised domain adaptation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 53–59, Uppsala, Sweden, July. Association for Computational Linguistics.

H. Daumé III. 2009. Frustratingly easy domain adaptation. *CoRR*, abs/0907.1815.

O. Dušek, J. Hajič, J. Hlaváčová, M. Novák, P. Pecina, R. Rosa, A. Tamchyna, Z. Urešová, and D. Zeman. 2014. Machine translation of medical texts in the Khresmoi project. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 221–228, Baltimore, MD, USA. Association for Computational Linguistics.

O. Dušek and F. Jurčíček. 2013. Robust Multilingual Statistical Morphological Generation Models. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*, pages 158–164, Sofia. Association for Computational Linguistics.

O. Dušek, Z. Žabokrtský, M. Popel, M. Majliš, M. Novák, and D. Mareček. 2012. Formemes in English-Czech deep syntactic MT. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, page 267–274.

O. Dušek, L. Gomes, M. Novák, M. Popel, and R. Rosa. 2015. New language pairs in TectoMT. In *Proceedings of WMT*. Under review.

M. Eck, S. Vogel, and A. Waibel. 2004. Language model adaptation for statistical machine translation based on information retrieval. In M. T. Lino, M. F. Xavier, F. Ferreira, R. Costa, and R. Silva, editors, *Proceedings of the International Conference on Language Resources and Evaluation*, pages 327–330, Lisbon, Portugal. European Language Resources Association.

A. S. Hildebrand, M. Eck, S. Vogel, and A. Waibel. 2005. Adaptation of the translation model for statistical machine translation based on information retrieval. In *Proceedings of the 10th Annual Conference of the European Association for Machine Translation*, pages 133–142, Budapest, Hungary. European Association for Machine Translation.

P. Koehn and J. 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. ACL.

P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

P. Langlais. 2002. Improving a general-purpose statistical translation engine by terminological lexicons. In *COLING-02 on COMPUTERM 2002: second international workshop on computational terminology*, volume 14, pages 1–7, Taipei, Taiwan. ACL.

L. Macken, J. Trushkina, and L. Rura. 2007. Dutch parallel corpus: MT corpus and translator's aid. In *Proceedings of the Machine Translation Summit XI*, pages 313–320. European Association for Machine Translation.

S. Mansour, J. Wuebker, and H. Ney. 2011. Combining translation and language model scoring for domain-specific data filtering. In *International Workshop on Spoken Language Translation*, pages 222–229, San Francisco, CA, USA. ISCA.

D. Mareček, M. Popel, and Z. Žabokrtský. 2010. Maximum entropy translation model in dependency-based mt framework. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 201–206, Uppsala, Sweden, July. Association for Computational Linguistics.

R. McDonald, F. Pereira, K. Ribarov, and J. Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530.

R. C. Moore and W. Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. ACL.

P. Nakov. 2008. Improving English–Spanish statistical machine translation: Experiments in domain adaptation, sentence paraphrasing, tokenization, and recasing. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 147–150, Columbus, OH, USA. ACL.

Martin Popel and Zdeněk Žabokrtský. 2010. TectoMT: Modular NLP framework. In Hrafn Loftsson, Eirikur Rögnvaldsson, and Sigrun Helgadottir, editors, *Lecture Notes in Artificial Intelligence, Proceedings of the 7th International Conference on Advances in Natural Language Processing (IceTAL 2010)*, volume 6233 of *Lecture Notes in Computer Science*, pages 293–304, Berlin / Heidelberg. Iceland Centre for Language Technology (ICLT), Springer.

G. Sanchis-Trilles and F. Casacuberta. 2010. Log-linear weight optimisation via Bayesian adaptation in statistical machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1077–1085, Beijing, China. ACL.

P. Sgall, E. Hajičová, and J. Panevová. 1986. *The meaning of the sentence in its semantic and pragmatic aspects*. D. Reidel, Dordrecht.

D. J. Spoustová, J. Hajič, J. Votrubec, P. Krbec, and P. Květoň. 2007. The Best of Two Worlds: Cooperation of Statistical and Rule-based Taggers for Czech. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies*, pages 67–74, Stroudsburg, PA, USA. Association for Computational Linguistics.

J. Straková, M. Straka, and J. Hajič. 2014. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18. Association for Computational Linguistics.

Z. Žabokrtský and M. Popel. 2009. Hidden Markov Tree Model in Dependency-based Machine Translation. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 145–148, Stroudsburg, PA, USA. Association for Computational Linguistics.

Z. Žabokrtský, J. Ptáček, and P. Pajas. 2008. TectoMT: highly modular MT system with tectogrammatics used as transfer layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 167–170. Association for Computational Linguistics.