Neural vs. Phrase-Based Machine Translation in a Multi-Domain Scenario

M. Amin Farajian[1,2], **Marco Turchi**[1], **Matteo Negri**[1], **Nicola Bertoldi**[1] **and Marcello Federico**[1]

[1]Fondazione Bruno Kessler, Human Language Technologies, Trento, Italy
[2]University of Trento, ICT Doctoral School, Trento, Italy
{farajian,turchi,negri,bertoldi,federico}@fbk.eu

Abstract

State-of-the-art neural machine translation (NMT) systems are generally trained on specific domains by carefully selecting the training sets and applying proper domain adaptation techniques. In this paper we consider the real world scenario in which the target domain is not predefined, hence the system should be able to translate text from multiple domains. We compare the performance of a generic NMT system and phrase-based statistical machine translation (PBMT) system by training them on a generic parallel corpus composed of data from different domains. Our results on multi-domain English-French data show that, in these realistic conditions, PBMT outperforms its neural counterpart. This raises the question: is NMT ready for deployment as a generic/multi-purpose MT backbone in real-world settings?

1 Introduction

Neural machine translation systems have recently outperformed their conventional statistical counterparts in the translation tasks in several domains such as news (Sennrich et al., 2016a), UN documents (Junczys-Dowmunt et al., 2016), and spoken language data (Luong and Manning, 2015). One common pattern in all these cases is that the target domain is always predefined, hence it is feasible to perform domain adaptation techniques in order to boost system performance for that particular application. However, in real-world applications it is very hard, if not impossible, to develop and maintain several specific MT systems for multiple domains. This is mostly due to the fact that usually: *i)* the target domain is not known in advance, and users might query different sentences from different domains; *ii)* the application domains are very diverse, which makes the possibility of developing and fine-tuning one system for each domain unfeasible; *iii)* there is no (or very limited amount of) in-domain training data to train domain-specific MT engines. In this situation, it is necessary to have high quality MT systems that perform consistently well in all (or most of) the domains. This problem becomes more important when we consider the case of small/mid-size language service providers, and their limited resources, which forces them to have few MT engines, but as much accurate as possible.

Considering the challenges posed by real-world applications, the recent NMT hype has hence to be put into perspective, trying to understand whether, in specific conditions, the neural paradigm is the Holy Grail for MT or not. To this aim, in this paper we compare the performance of phrase-based SMT (PBMT) and neural MT (NMT) systems in a real-world scenario in which the systems are trained on a combination of multiple domains, and analyse their differences and behaviours. Our experiments on an English-French data set, suggest that there is still some way to go to make NMT really usable "into the wild" (*i.e.* to make it stable and robust to multi-domain training data). In Section 2 we review the state-of-the-art approaches of multi-domain machine translation for both PBMT and NMT. In Section 3 we describe our experimental setup. The results are described and analysed in Section 4, where we compare different behaviours of PBMT and NMT in more details.

2 Multi-Domain Machine Translation

Multi-domain machine translation is very well-studied in the field of statistical phrase-based MT. The approaches proposed for this issue vary from learning a single model from *pooled* training data,

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 280–284,
Valencia, Spain, April 3-7, 2017. ©2017 Association for Computational Linguistics

to more complicated (log-)linear interpolations of multiple models using mixture models (Foster and Kuhn, 2007) and linear mixture models (Carpuat et al., 2014).

However, being a very new field of research, to the best of our knowledge, there is no work on developing multi-domain NMT systems. However, to the best of our knowledge, there is still no work on developing multi-domain systems (*i.e.* generic/multi-purpose systems trained with all the data available at a given time) within the state-of-the-art NMT framework. Indeed, though interesting and well motivated from an application-oriented perspective (*e.g.* think about a translation company looking for a generic MT backbone usable for jobs coming from any domain), this issue is still unexplored. The current state-of-the-art research in NMT explored the effectiveness of domain adaptation, and the approaches for how to adapt existing NMT systems to a new domain (Luong and Manning, 2015). The assumption of these works, however, is that the new target domains are either known in advance or presented together after some sample data have been made available to fine-tune the system. There exist an active field of research that is trying to solve a quite different issue that has a similar motivation, which is multi-lingual NMT (Firat et al., 2016a; Firat et al., 2016b; Johnson et al., 2016). The motivations behind these works are very similar to the ones described in Section 1, which is mostly simplifying the deployment of MT engines in the production lines. So, the final goal is to reduce the number of final systems, trained with pooled multi-domain data sets, without degrading the final performance. As we will see in the remainder of this paper, this issue is still open, especially when we embrace the state-of-the-art NMT paradigm.

3 Experimental Setup

3.1 Data

To mimic the real-world applications, we trained our generic systems on a collection of publicly available English-French data from different domains: European Central Bank (ECB), Gnome, JRC-Acquis (JRC), KDE, OpenOffice (OOffice), PHP, Ubuntu, and translated UN documents (UN-TM).[1] Since the size of these corpora are relatively small for training robust data-driven MT systems, in particular NMT solutions, we used CommonCrawl and Europarl corpora as out-domain data in addition to the above-mentioned domain-specific corpora, resulting in a parallel corpus of 5.5M sentence pairs. The statistics of the corpora are presented in Table 1. All the corpora are pre-processed by normalizing punctuation, removing special characters, tokenizing, truecasing, and removing empty lines as well as sentences with lengths greater than 50 and also the ones with length ratio greater than (1:9), using the standard Moses scripts. Then, a set of 500 sentence pairs from each domain is selected randomly as development and 1000 sentence pairs as held-out test corpus; duplicated sentence pairs are then removed from each corpus separately, resulting in a total of 3,527 and 6,962 sentence pairs for dev and test corpora for all the domains. The statistics of the test corpora are reported in Table 2.

	Segments	**Tokens**	**Types**
ECB	147.7K	3.1M	40.9K
Gnome	238.4K	1.7M	16.8K
JRC	689.2K	10.8M	78.4K
KDE4	163.2K	1.0M	42.0K
OOffice	34.5K	389.0K	9.3K
PHP	38.4K	259.0K	9.7K
Ubuntu	9.0K	47.7K	8.6K
UN-TM	40.3K	913.8K	12.5K
CommonCrawl	2.6M	57.8M	759.4K
Europarl	1.7M	39.6M	111.0K

Table 1: Statistics of the English side of the original corpora, after pre-processing.

	Segments	**Tokens**	**Types**
ECB	1000	20.9K	3.8K
Gnome	982	7.3K	1.9K
JRC	757	14.8K	2.9K
KDE4	988	14.8K	2.1K
OOffice	976	11.1K	1.9K
PHP	352	5.3K	1.3K
Ubuntu	997	5.1K	1.9K
UN-TM	910	22.2K	3.1K

Table 2: Statistics of the English side of the test corpora.

3.2 Phrase-based SMT

The experiments of the phrase-based SMT systems are carried out using the open source Moses

[1] All these corpora are available in http://opus.lingfil.uu.se

toolkit (Koehn et al., 2007). The word alignment models are trained using fast-align (Dyer et al., 2013). In our experiments we used 5-gram language models trained with modified Kneser-Ney smoothing using KenLM toolkit (Heafield et al., 2013). The weights of the parameters are tuned with batch MIRA (Cherry and Foster, 2012) to maximize BLEU on the development set. Development set is a combination of all the development corpora of all the domains.

3.3 Neural MT

All the experiments of the NMT systems are conducted with the Nematus toolkit[2] which is an implementation of the attentional encoder-decoder architecture (Bahdanau et al., 2014). Since handling large vocabularies is one of the main bottlenecks of the existing NMT systems, in practice the state-of-the-art NMT systems are trained on the training corpora in which the less frequent words are segmented into their sub-word units (Sennrich et al., 2016b) by applying the modified version of the byte pair encoding (BPE) compression algorithm (Gage, 1994). This makes the NMT systems capable of dealing with new and rare words, resulting in open-vocabulary translations. Following the common practice in the field, we segmented the training corpora using the scripts provided by the Nematus toolkit. As recommended by (Sennrich et al., 2016b), in order to increase the consistency in segmenting the source and target text, the source and target side of the training set are combined and number of merge rules is set to 89,500, resulting in vocabularies of size 78K and 86K tokens for English and French languages, respectively. We use mini-batches of size 100, word embeddings of size 500, and hidden layers of size 1024. The maximum sentence length is set to 50 in our experiments. The models are trained using Adagrad (Duchi et al., 2011), reshuffling the training corpora for each epoch. The models are evaluated every 10,000 mini-batches via BLEU (Papineni et al., 2002). It is worth mentioning that with the same set-up we recently achieved state-of-the-art performance in the International Workshop on Spoken Language Translation evaluation (Farajian et al., 2016).

4 Analysis and Discussion

Table 3 presents the results of the generic systems (*PBMT gen.* and *NMT gen.*) and the NMT system adapted to the concatenation of all the eight specific domains (*NMT-adp.jnt*), as well as the NMT systems which are specifically adapted to each domain separately (*NMT-adp.sep*). In the case of *NMT-adp.jnt* and *NMT-adp.sep* we used the best model of the *NMT gen.* and adapted it to their corresponding training corpora by continuing the training for several epochs, using the training data of that specific domain.

4.1 NMT vs. PBMT in Multi-domain scenario

As the results show, the generic PBMT system outperforms its NMT counterpart in all the domains by a very large margin; and as the NMT system becomes more specific by observing more domain-specific data, the gap between the performances reduces until the NMT outperforms; which confirms the results of the previous works in this field (Luong and Manning, 2015). However, it is interesting to see what is the reason behind the very low performance of the generic NMT system compared to the generic PBMT. First, we noticed that in the case of PHP corpus, the text is very noisy (*ie.* misaligned sentences) which makes it hard for the system to learn reliably. For instance, we observed that in one case, the same English sentence is aligned with more than 20 French sentneces which are mostly wrong translations.

Second, by analysing the number of repeated sentence pairs in the training corpora we observed that Gnome corpus has the highest repetition rate among all the domains (each sentence is repeated 4.6 times in average), hence leaving a large space for NMT to memorize the translation patterns of this specific domain. This can partially justify the reason behind the very large gain after adapting the NMT system in this domain.

Third, we noticed that in the case of Ubuntu domain, the gain of domain adaptation is very minimal for both of the adapted NMT systems. By looking at the *token/type ratios* we observed that this specific domain has the lowest ratio, 5.12, which means each word is observed around 5 times in the corpus, while for the other corpora is at least five times more; ranging from 25.35 in the case of KDE corpus to 146.34 in the case of JRC-Acquis. In our opinion there is a high rela-

[2] `https://github.com/rsennrich/nematus`

	PBMT gen.	NMT gen.	NMT adp. jnt.	NMT adp. sep.
Overall	61.06	48.25	54.67	62.32
ECB	58.61	46.53	52.23	58.04
Gnome	90.54	61.49	79.26	93.76
JRC	66.26	56.49	61.00	62.62
KDE4	50.64	46.36	51.29	55.71
OOffice	37.11	31.75	35.45	39.85
PHP	47.04	33.43	34.23	39.73
Ubuntu	45.76	45.27	46.14	46.87
UN-TM	69.69	52.14	60.53	75.72

Table 3: Performance of the generic and adapted systems in terms of BLEU score.

tion between the token/type ratio and the amount of gain obtained in the domain adaptation phase.

4.2 Open Vocabulary Translation in Technical Domains

The word segmentation approach proposed in (Sennrich et al., 2016b) has been shown to be very effective in obtaining open vocabulary translation with a fixed vocabulary in NMT. While this holds true for several cases such as morphologically complex words, we noticed that in more technical domains where the text contains technical words and terms, such as application names, splitting the words into multiple tokens can make the translation harder for the NMT systems. In many of these cases we observed that the human translators prefer not to translate the term and use them as they are. In these cases, the PBMT system that copies the unknown words into the output is rewarded, while the NMT system often misses the proper translation of at least one sub-word unit, resulting in a wrong translation of the full word. For example, let's consider the out-of-vocabulary word `Bluetile`, which belongs to the Ubuntu domain but was not seen during training. The PBMT system copies the word in the output while the NMT system segments it to `Blu@@`, `eti@@`, and `le` and translates them into `Blu@@`, `et@@`, and `le`, resulting in `Bluetle`.

Another interesting phenomenon that we observed is that in some cases the NMT system translates the sub-word units properly, while in that context the word should not be translated and copied in the target sentence as it is. For instance, the following sentence which belongs to the Ubuntu manual is just describing the usage of an application and its corresponding options, hence the switches should not be translated:

```
-D, --disconnect disconnect
```

In this case the token `--disconnect` is unknown to both systems. The PBMT system as described earlier copies the token, while NMT first segments the token into `--@@` and `disconnect`, and then translates them as `--@@` and `deconnexion`, respectively.

These cases show that while sub-words obtained by applying BPE are crucial to obtain open vocabulary translation in generic domains, one should be very careful in applying them in specific domains containing large number of technical terms.

4.3 Is NMT Ready for Deployment?

Recently, (Junczys-Dowmunt et al., 2016) performed a very extensive experiment in which the performance of NMT is compared with PBMT and hierarchical SMT on multiple language directions and showed that NMT systems in almost all the cases outperform their SMT counterparts and to solve the only remaining issue which is the decoding time of the NMT systems, they introduce an efficient neural decoder which makes it feasible to deploy NMT systems in-production line. However, all their experiments are performed on one single domain for which there exists a very large training corpus.

In our experiment, we observed that the generic NMT systems are by a large margin behind their PBMT counterparts in the real-world scenarios (48.25 versus 61.06 BLEU score) where the training data are very heterogeneous and are composed of multiple corpora with different sizes (varying from very few thousands to millions of sentence pairs). This suggests that in order to be deployed in production lines, NMT systems need to be armed with more efficient mechanisms, which enables them to deal with more heterogeneous data.

5 Conclusion

In this paper we studied the capability of neural machine translation systems in the real-world applications were the training corpora consist of text obtained from different domains; and compared them with their phrase-based counterparts. Our results on multi-domain English-French data showed that, in these realistic conditions, PBMT outperforms NMT by a large margin.

Acknowledgments

This work has been partially supported by the EC-funded H2020 projects QT21 (grant no. 645452) and ModernMT (grant no. 645487).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Marine Carpuat, Cyril Goutte, and George Foster. 2014. Linear mixture models for robust machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 499–509, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 427–436, Stroudsburg, PA, USA. Association for Computational Linguistics.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research.*

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *In Proc. NAACL.*

Amin M. Farajian, Rajen Chatterjee, Costanza Conforti, Shahab Jalalvand, Mattia A. Di Gangi, Duygu Ataman, Marco Turchi, Negri Matteo, and Marcello Federico. 2016. Fbks neural machine translation systems for iwslt 2016. In *Proceedings of the International Workshop on Spoken Language Translation*, Seattle, US, December.

Orhan Firat, KyungHyun Cho, and Yoshua Bengio. 2016a. Multi-way, multilingual neural machine translation with a shared attention mechanism. *CoRR*, abs/1601.01073.

Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016b. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas, November. Association for Computational Linguistics.

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.

Philip Gage. 1994. A New Algorithm for Data Compression. *C Users J.*, 12(2):23–38, February.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified kneser-ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696, Sofia, Bulgaria, August. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. 2016. Is neural machine translation ready for deployment? a case study on 30 translation directions. In *Arxiv*, October.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford Neural Machine Translation Systems for Spoken Language Domains. In *Proceedings of the International Workshop on Spoken Language Translation.*

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, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh Neural Machine Translation Systems for WMT 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany, August. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics.