Extracting and Selecting Relevant Corpora for Domain Adaptation in MT

Lars Bungum
Norwegian University of Science and Technology
Dept. of Computer and Information Science
Sem Sælands vei 7
N-7490 Trondheim, Norway
larsbun@idi.ntnu.no

Abstract

The paper presents scheme for doing Domain Adaptation for multiple domains simultaneously. The proposed method segments a large corpus into various parts using self-organizing maps (SOMs). After a SOM is drawn over the documents, an agglomerative clustering algorithm determines how many clusters the text collection comprised. This means that the clustering process is unsupervised, although choices are made about cut-offs for the document representations used in the SOM.

Language models aren then built over these clusters, and used as features while decoding a Statistical Machine Translation system. For each input document the appropriate auxiliary Language Model most fitting for the domain is chosen according to a perplexity criterion, providing an additional feature in the log-linear model used by Moses. In this way, a corpus induced by an unsupervised method is implemented in a machine translation pipeline, boosting overall performance in an end-to-end experiment.

1 Introduction

Broadly viewed, the problem of Domain Adaptation (DA) is relevant to many computer applications, not only Artificial Intelligence (AI) and Machine Translation in (MT), the focus of this research. A MT system fit for a specific or generic domain or purpose, often has trouble is translating text from another domain, consisting of different input data, a claim backed by Carpuat et al. (2012).

To illustrate, suppose a MT system trained on- or tuned for a certain text domain, say archery, is used on a text about a completely different field such as string quartets. One would, among other things, assume that the meaning of the word *bow* is different in the two domains, and is likely to require different translations into some other languages. (Differences in text domains can also pertain to other features of language, such as punctuation and grammar.) Furthermore, the assumption is that a small in-domain system is already built, and the characteristics of this model and its training data are used for selecting more domain specific text from a larger, general source.

Alternatively, a system is built on a general corpus, and needs to be adapted to domain-specific text as it comes in. In this paper we will present a method that allows for a general-purpose MT system to be adapted to multiple domains online. This is achieved by using an unsupervised method to cluster unorganized text into segments and combining Language Models (LMs) built on these segments via log-linear feature functions. As new text is input to the MT system, an assessment is done at the document level to select the appropriate domain-specific LM.

In Statistical Machine Translation (SMT) contexts, data-driven MT systems are trained on parallel and monolingual training data. When in need of translating text belonging to a certain domain, domain specific training material is often hard to come by, whereas general (other) text exists in abundance. Using the web as a corpus has an obvious appeal, as Kilgarriff and Grefenstette (2003) effectively demonstrated, but since data from the web is usually not structured, effectively making use of this knowledge source is difficult. While finding more *in*-domain monolingual text is easier than bi-lingual text, it is still not trivial.

When faced with a translation task where the training material for a specific domain to be translated (the *in*-domain) is scarce, one answer to the problem is using refined machine learning methodologies such as active learning (see Rai et

D S Sharma, R Sangal and J D Pawar. Proc. of the 11th Intl. Conference on Natural Language Processing, pages 336–343,
Goa, India. December 2014.

al. (2010) for a discussion of Active Learning and Domain Adaptation) to exploit the (often little) domain specific training material available and building a new SMT model trained on it. Another approach is to use scarce in-domain data as a starting point to collect more in-domain training material with bootstrapping methods, similar to the little one begins with, from a larger source such as the Internet. It is possible to expand the available *in*-domain data, be it mono- or bi-lingual text, and to use the *in*-domain data more efficiently, or both. (See Wu et al. (2009) who employ bootstrapping for Domain Adaptation in a Named-Entity Recognition task).

We employed an unsupervised algorithm, the Self-Organizing Map (SOM), to create order in an otherwise unorganized body of text and used this to create auxiliary Language Models (LMs) for SMT decoding. Decisions had to be taken on how the documents were represented as vectors, and we used an agglomerative algorithm to decide how many clusters should be created from the SOM with bottom-up hierarchical clustering.

The algorithm provides n separate clusters of text, from which standard n-gram LMs were built. With this method, the number of auxiliary text corpora (and later LMs) are determined by the agglomerative clustering algorith, enabling Domain Adaptation into the *available* domains, which is why an unsupervised method was chosen. The LMs were used in a SMT pipeline (Moses), implemented as features in Moses' log-linear decoder. When a document is input for translation, it was matched against the n LMs created above, ranked after perplexity. The LM with the lowest perplexity was selected by the Moses feature to provide additional information for the decoder. This setup creates a platform in which a system can do adaption to multiple domains, as the additional feature in the SMT decoding phase can select the most appropriate auxiliary LM on-the-fly. Additionally the SOM-approach to Domain Adaptation is evaluated in an end-to-end MT context, although with rudimentary evaluation on just one dataset.

2 Technical Overview

The implementation consists of two stages, (i) the segmentation of a large corpus with a SOM and (ii) the utilization of language models built on the basis of these corpus segments in an SMT system . The first phase is conducted offline, whereas the employment of the language models is done while decoding a SMT model given input sentences.

The steps in the offline and online parts of the system are summarized in Figure 1. Once the first offline phase is completed, the system is able to adapt to any number of incoming text domains, via the Moses feature that selects the most appropriate LM depending for the input document based on a perplexity measure.

2.1 SOM-Induced Corpora

When Kohonen et al. (1996) introduced Self-Organizing Maps (SOMs) they were used to cluster USENET (newsgroup) data. Later, the same methodology was used to cluster patent data (Kohonen et al., 2000) and the Encyclopedia Britannica (Lagus et al., 2004).

SOMs are maps of vectorized data that can cluster high-dimensional data within a lower (most often 2 or 3-dimensional) topology. It is a way of showing similarities between high-dimensional data (such as documents represented with tens of thousands of dimensions) in a low-dimensional space. The algorithm is summarized in the following steps:

1. Create n random nodes in the low-dimensional map according to some topology.
2. Pick an input (document) vector.
3. Associate with the node that is closest according to a distance measure.
4. Update the node and its neighbors to be more like the vector.

The corpus used in the experiments was the SdeWac (Faa and Eckart, 2013) corpus of German text, consisting of parsable sentences. Each document in the collection was vectorized with Scikit-learn (Pedregosa et al., 2011), which has many different vectorizers available, such as TF-IDF and N-gram vectorizers, on word and character level. In these experiments, a mostly unigram TF-IDF vectorizer was used, some tests were also run using it in combination with N-gram frequencies and for bigrams.

The SOM algorithm was implemented in MPI (mpi4py) and Python, and run on a PBS scheduler. This means that no changes to the SOM algorithm as presented by Kohonen (2001) were needed. A

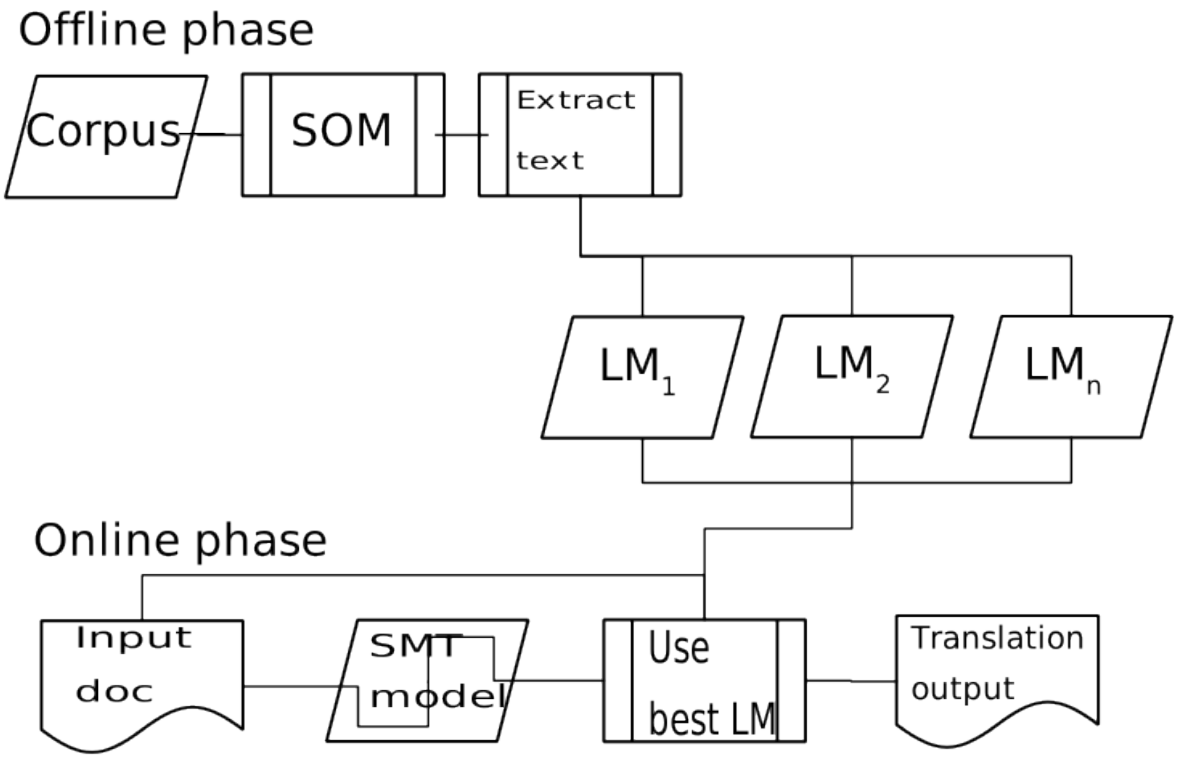


Figure 1: Overall architecture

quadratic layout of nodes was used in the experiments, and for each sample, the comparisons with all the different vectors in the node were done in parallel, as was the updating of the nodes in the next stage. But the traversal of the samples was done serially, so that the SOM was updated according to the information supplied in each sample after it had found its winning node.

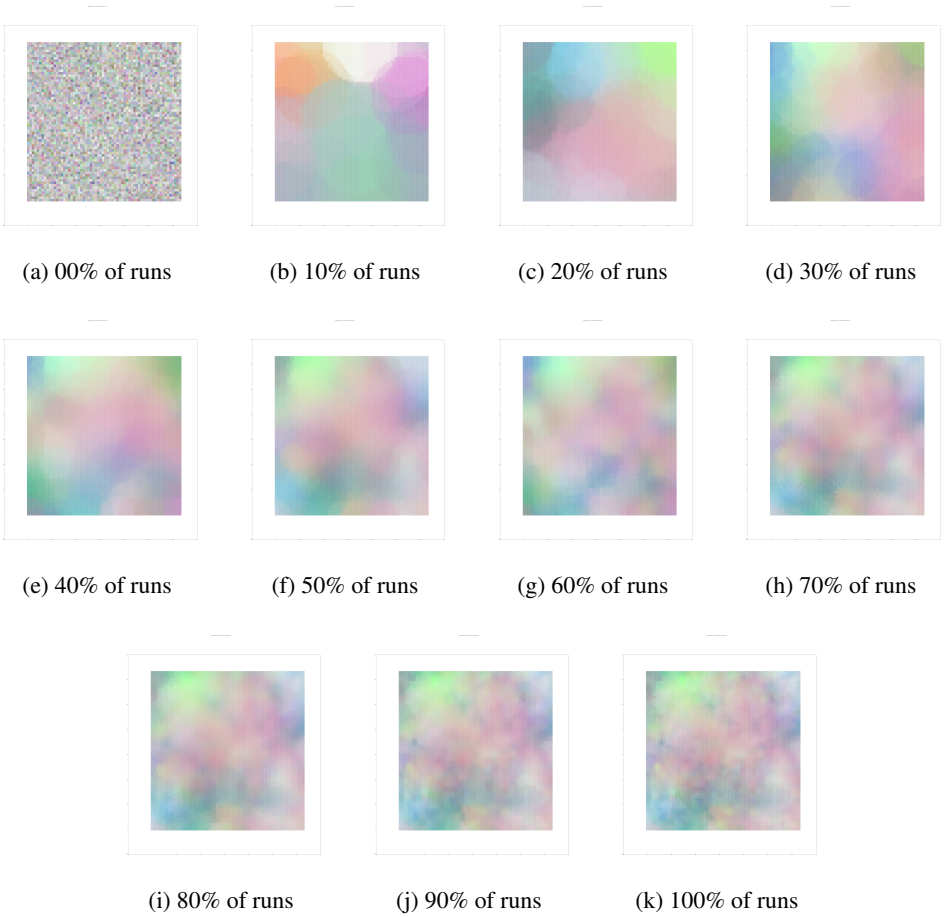
After a maximum number of iterations was reached (usually set to 100 or 1000), the last run of the input samples was kept, which left similar samples with the same winning node. An agglomerative clustering algorithm was then run on the nodes to cluster them by similarity. The algorithm chose a cut-off point in the clustering according to the relative change in the similarity measure.

The development of the SOM from its randomized beginning is displayed in Figure 2. Each of the nodes contains a vector of the dimensionality printed in the caption. These vectors are then scaled down to four dimensions with Principal Component Analysis (Jolliffe, 2005), that Matplotlib (Hunter, 2007) allows for as input to its color printing (four dimensions were preferred over three to allow for one more component). It can be seen on the figures that areas of similar documents (as indicated by reducing to about the same color) arise towards the end of the cycle.

2.2 Selection of Relevant Corpora

Finally, each of the *n* clusters resulting from the agglomerative clustering had a certain number of samples belonging to them. The samples (text documents) were then output and placed in the same directory. When the files were concatenated they were again text corpora over which normal n-gram models were constructed.

Hierarchical agglomerative clustering algorithms are algorithms that successively merge clusters from the bottom up. Whereas the top-down approach would need metrics to split clusters, the bottom-up approach needs criteria to merge clusters. We used the algorithms provided by the Scipy package (Jones et al., 2001). Running the algorithm, a distance metric to determine similarity between clusters is picked, as well as a selection of which data points to measure distances between. Distance metrics, for instance Euclidean or Chebychev distances, are combined with a a choice of nodes in the grid to calculate distance between to determine the distance between clusters (comprised of more and more nodes in the



(a) 00% of runs (b) 10% of runs (c) 20% of runs (d) 30% of runs

(e) 40% of runs (f) 50% of runs (g) 60% of runs (h) 70% of runs

(i) 80% of runs (j) 90% of runs (k) 100% of runs

Figure 2: Development of SOM with 77,733 dimensions, 4096 nodes and 100 iterations

agglomerative algorithm). The average distance was used in these experiments, the distance between the closest, or the most separated once are alternatives. A dendrogram of the agglomerative clustering is shown in Figure 3, next to a plot of the distance between the clusters on the vertical axis against the possible clusterings on the horizontal. The number of clusters was chosen by finding a knee-point in this curve, where the marginal gain of adding another cluster is the highest. (This is hard to interpret visually on a curve with this many points.)

2.3 Multiple Language Models as a Feature in Moses

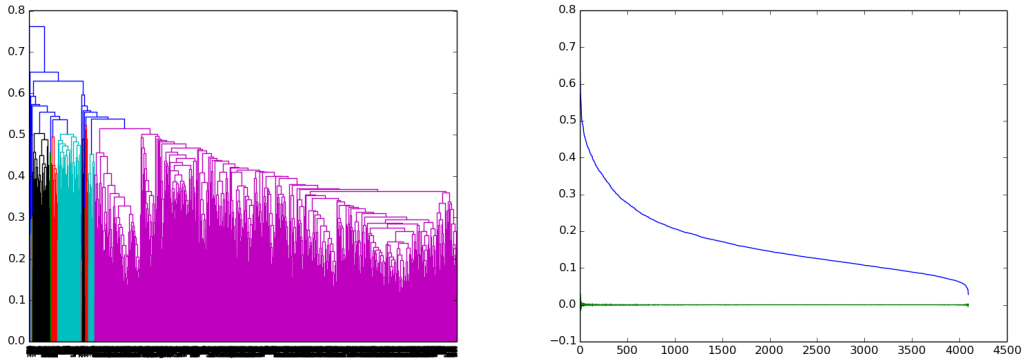
The Moses (Koehn et al., 2007) SMT system allows for the use of user-defined features in its log-linear model. A feature was created using one of the SOM-induced LMs created in the above step for scoring the sentences, depending on what LM gave the lowest perplexity score to the document that any given input sentence belonged to. As each SL sentence was read, it would leave a number designating which LM to use for the feature in decoding its translation.

This way, information at the document level provided information about which LM is the best to use for the documents as they come in. The SRILM kit was used to create the LMs and using corresponding libraries were used to score hypotheses inside the Moses feature.

3 Preliminary Results

A preliminary evaluation was done on sentence-level WMT12 data as proof of concept. An SMT model was trained on the Europarl (Koehn, 2005) and News Commentary[1] corpora combined and tested on the newstest2012 dataset. One of the LMs generated with the method presented in Section 2.1 was used for the entire test set. After hav-

[1]The WMT News Commentary parallel corpus contains news text and commentaries from the Project Syndicate and is provides as training data for the series of WMT translation shared tasks (See `http://statmt.org/`).



(a) Dendrogram of clusters

(b) Agglomerative clustering of SOM

Figure 3: Illustrations of how the SOM map is clustered

ing used the newstest2011 data for MERT (Och, 2003) tuning of the feature weights, a gain of 2.44 BLEU (Papineni et al., 2002) points was achieved when using the feature presented in Section 2.3.

The full implementation is yet to be evaluated, as we are still working on preparing corpora that can be processed at the document level. This can be done by processing the newstest corpora provided in the WMT datasets as the provided *sgml* files are included with document IDs, making it possible to identify separate documents, unlike some other available parallel corpora. This is also the case for the biography dataset created by Louis and Webber (2014).

4 Related Research

Domain Adaptation generalizes beyond the scope of Machine Translation, and can be viewed as a general Machine Learning task. Blitzer (2008) performed a rigorous analysis of Domain Adaptation algorithms and under what conditions they perform well, conducting experiments in sentiment classification and part-of-speech (POS) tagging. Hildebrand et al. (2005) employed information retrieval techniques for DA in MT by selecting the sentences in the training material that were similar to the ones used in the test set.

Xu et al. (2007) built separate SMT models based on little *in*-domain data for building both translation and language models. The models were used in combination with a larger, general SMT system, as the *in*-domain training data was limited. The authors performed DA as a classification task, where a document is classified before translation to belong to a certain domain, and the corresponding SMT model was used. The authors found that classifying the right domain for an input document was more accurate when measuring their perplexity on a Language Model than other, Information Retrieval-based methods.

While their work mainly focused on adapting translation models, Sennrich et al. (2013) also investigated the idea of using unsupervised methods to classify text, combining them with mixture models to perform Domain Adaptation. Louis and Webber (2014) used cache models to store domain specific information in language modeling, also implemented as features in the Moses SMT system, that was also used in this work.

Moore and Lewis (2010) used a perplexity criterion to select the best corpus for building an auxiliary LM by testing the available extra data on the sentence level to extract the relevant parts at the sentence level according to a cross-entropy threshold. They showed that it is not necessary to use the whole additional text in order to obtain improvement in performance. Building on this idea, our work also uses only segments of the total corpus on which the SOM is drawn, but selects the auxiliary LM based on the perplexity of the *input* document at decode time.

Axelrod et al. (2011) did a similar extraction of what they term *pseudo-in-domain* sentences based on cross-entropy measures; pseudo because they are similar, but not identical to the in-domain data. Their measures of perplexity and cross-entropy is also done at sentence level. They demonstrated increased performance in an end-to-end experiment using only parts of a large, general corpus. With the firepower of modern computers, how-

ever, it is well feasible to build large LMs efficiently ((Bungum and Gambäck, 2012)) on quite standard equipment, so a comparison with the performance of the entire general corpus on the end-to-end task would be interesting.

5 Discussion

The methodology in the present work stipulates a solution to the Domain Adaptation problem that looks for external sources to increase the available training data. The SOM approach is a way of finding similarities in unorganized data collections that has been applied successfully in other application areas. This is the first stage in creating separate language models from a large web corpus, to aid translation of a specific language domain.

In Machine Translation history there are several accounts of systems working well for a specific domain, but very hard to build a system working in any domain (general-purpose) while retaining high quality. Since it is feasible to translate text for one specific domain well, but very hard to translate general text with the same high quality, bridging the gap between these two processes is a possible way forward.

The work presented here needs to be tested more rigorously, as it is hard to find datasets consisting of many different domains, sorted on document level that are needed to test the idea fully. Otherwise, while the scale of the project is large, and it requires significant computer resources, it is still well within what it is possible to incorporate into one SMT model, simply by adding the corpus that we are doing clustering on to the training material. In principle though, given the scale of the Internet and the growth of content added, it is not possible to add all new text that can be crawled from online resources in any system, and some sort of segmentation is desirable.

Our approach builds on earlier work in segmenting a large corpus into relevant parts and using this to aid the overall MT task. Relating to other work on DA it presents a method where a general MT system can be adapted to multiple target domains at the same time. As discussed in (Bungum and Gambäck, 2011) it is not always obvious how to separate text domains from each other, where to draw the line between them, and what dimensions (such as writing style, topic, author or target age groups) through which to separate them. Using an unsupervised approach segmentation of a vast data source is a way of enabling a MT system to respond to various input domains also along such dimensions.

6 Future Work

Looking forward, we would like to test this method on more languages and more parallel corpora to see if it generalizes well. There is also extensive literature on language domains and sublanguages as they can be characterized not only by thematic variance and genre, but also differences in the properties of the author (age, style, emotional state). It is not obvious that this method works equally well for all such situations.

The method proposed here integrates many parts in the two stages of the process. Especially in the SOM step there are many choices to be made regarding how documents are represented, and how the number of clusters are chosen as the nodes are joined together in the final step. The system is implemented with Scikit-learn so that alternative similarity measures both in running vector comparisons in the SOM and cluster similarity can be used. In many runs the resulting text corpora varied greatly in size with one big corpus dominating the others. Trying to skew the agglomeration towards more equal-sized partitions is an interesting avenue to pursue.

There has also been interesting work on trying to mine more text based on a little in-domain corpus from the web. Such approaches could also be integrated in these experiments by using quantitative data on the in-domain corpus to compute vector representations. Finally, extending the SOM approach to also mine parallel and not just monolingual corpora is a goal that can further advance Machine Translation performance. More monolingual data certainly helps, but more high-quality parallel text would arguably help even more.

Acknowledgements

I thank Björn Gambäck for valuable feedback on this article.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 355–362, Edinburgh, United Kingdom. Association for Computational Linguistics.

John Blitzer. 2008. *Domain Adaptation of Natural Language Processing Systems*. Ph.D. thesis, University of Pennsylvania.

Lars Bungum and Björn Gambäck. 2011. A survey of domain adaptation in machine translation: Towards a refinement of domain space. In *Proceedings of the India-Norway Workshop on Web Concepts and Technologies*, Trondheim, Norway. Tapir Academic Press.

Lars Bungum and Björn Gambäck. 2012. Efficient N-gram Language Modeling for Billion Word Web-Corpora. Workshop on Challenges in the Managment of Large Corpora, LREC 2012.

Marine Carpuat, Hal Daumé III, Alexander Fraser, Chris Quirk, Fabienne Braune, Ann Clifton, Ann Irvine, Jagadeesh Jagarlamudi, John Morgan, Majid Razmara, Aleš Tamchyna, Katharine Henry, and Rachel Rudinger. 2012. Domain adaptation in machine translation: Final report. In *2012 Johns Hopkins Summer Workshop Final Report*.

Gertrud Faa and Kerstin Eckart. 2013. Sdewac – a corpus of parsable sentences from the web. In Iryna Gurevych, Chris Biemann, and Torsten Zesch, editors, *Language Processing and Knowledge in the Web*, volume 8105 of *Lecture Notes in Computer Science*, pages 61–68. Springer Berlin Heidelberg.

Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex 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, May. European Association for Machine Translation.

J. D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95.

Ian Jolliffe. 2005. *Principal component analysis*. Wiley Online Library.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2001–. SciPy: Open source scientific tools for Python. [Online; accessed 2014-07-17].

Adam Kilgarriff and Gregory Grefenstette. 2003. Introduction to the special issue on the web as corpus. *Comput. Linguist.*, 29(3):333–347, September.

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.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT.

Teuvo Kohonen, Samuel Kaski, Krista Lagus, and Timo Honkela. 1996. Very large two-level SOM for the browsing of newsgroups. In C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, and B. Sendhoff, editors, *Proceedings of ICANN96, International Conference on Artificial Neural Networks, Bochum, Germany, July 16-19, 1996*, Lecture Notes in Computer Science, vol. 1112, pages 269–274. Springer, Berlin.

Teuvo Kohonen, Samuel Kaski, Krista Lagus, Jarkko Salojrvi, Vesa Paatero, and Antti Saarela. 2000. Organization of a massive document collection. *IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery*, 11(3):574–585, May.

Teuvo Kohonen. 2001. *Self-organizing maps*. Springer series in information sciences, 30. Springer, Berlin, 3rd edition, December.

Krista Lagus, Samuel Kaski, and Teuvo Kohonen. 2004. Mining massive document collections by the WEBSOM method. *Information Sciences*, 163(1-3):135–156.

Annie Louis and Bonnie Webber. 2014. Structured and unstructured cache models for smt domain adaptation. In Israel Shuly Wintner, University of Haifa, Germany Stefan Riezler, Heidelberg University, and UK Sharon Goldwater, University of Edinburgh, editors, *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*. Association for Computational Linguistics, April.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden, July. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.

K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, July. ACL.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Piyush Rai, Avishek Saha, Hal Daumé III, and Suresh Venkatasubramanian. 2010. Domain adaptation meets active learning. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, pages 27–32. Association for Computational Linguistics.

Rico Sennrich, Holger Schwenk, and Walid Aransa. 2013. A multi-domain translation model framework for statistical machine translation. In *ACL (1)*, pages 832–840. The Association for Computer Linguistics.

Dan Wu, Wee Sun Lee, Nan Ye, and Hai Leong Chieu. 2009. Domain adaptive bootstrapping for named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1523–1532, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jia Xu, Yonggang Deng, Yuqing Gao, and Hermann Ney. 2007. Domain dependent statistical machine translation. In *In Proceedings of the MT Summit XI*, pages 515–520.