Multi-Domain Adapted Machine Translation Using Unsupervised Text Clustering

Lars Bungum and Björn Gambäck

Department of Computer and Information Science
Norwegian University of Science and Technology
Sem Sælands vei 7–9, NO—7094 Trondheim, Norway
{larsbun,gamback}@idi.ntnu.no
http://www.idi.ntnu.no

Abstract. Domain Adaptation in Machine Translation means to take a machine translation system that is restricted to work in a specific context and to enable the system to translate text from a different domain. The paper presents a two-step domain adaptation strategy, by first making use of unlabeled training material through an unsupervised algorithm, the Self-Organizing Map, to create auxiliary language models, and then to include these models dynamically in a machine translation pipeline.

Keywords: (Statistical) Machine Translation, Domain Adaptation, Self-Organizing Maps, Hierarchical Clustering, Unstructured Data

1 Introduction

Intuitively we reply “it depends” when asked to translate a word into our native language. Indeed the meaning of a word depends, but on what? Research into Machine Translation (MT) is partially about making these dependencies tangible and accounted for. The dependencies in “it depends” can be seen as constraints on an MT system. Kay (1980) formulated these constraints as *quality*, *automation*, *general purpose* and *computational efficiency*. While no system has been even close to solving the unconstrained problem, Machine Translation is successful in scenarios where some of these requirements are relaxed. Restricting the system to certain contexts, by only allowing the input text to belong to a specific *domain* means relaxing the *general purpose* constraint. Domain Adaptation (DA) in turn means adapting an MT system catered for a general or specific purpose into a system capable of translating text from a different domain.

Faced with data from new domains and contexts, MT systems have problems, to a varying degree. Reasonably so, as some domains are closer than others, even if it can be hard to specify exactly what separates them: as language is constantly changing and evolving, it is difficult to draw boundaries between domains. When the focus of Machine Translation shifted from rule-based to data-driven (and later hybrid) approaches in the wake of the landmark IBM Models (Brown et al., 1990), Domain Adaptation received more attention as a particular problem. In data-driven approaches, training material is a scarce resource; hence all available data is used, without necessarily being adapted to any specific domain.

Plank (2011) and Jiang (2008) argued that annotating more data in the new domain is a theoretical, but not practical strategy, thus stating the goal of Domain Adaptation as making systems usable in new contexts without needing to annotate new data or to specifically exploit labeled data from one domain to train classifiers on a new domain. Jiang (2008) also stressed a major point when she observed that context and domain adaptation is a general problem, and that techniques developed for Domain Adaptation are applicable to most classification tasks where the distributions of the training and the test sets differ.

Here we aim to explore how a Machine Translation system could a) adapt to multiple domains without knowing the domain of the input document, and b) choose the appropriate domain on the basis of an entire input document (and not only one string). This would presuppose a system that can make use of, in principle, unlimited unsorted data, and furthermore could pre-process the data in such a way that it could be used online. Building on an idea from Moore and Lewis (2010), this entails addressing the question of how to segment a large text corpus into sections that can be used for Domain Adaptation. More specifically, the present work uses only segments of the total corpus on which a Self-Organizing Map is drawn, but selects an auxiliary Language Model based on the perplexity of the *input* document (i.e., roughly its level of ambiguity).

The rest of the paper is organised as follows: Section 2 discusses the background and related work on Domain Adaptation; Section 3 then introduces the methods and data sets used in the present work. In Section 4, Self-Organizing Maps are used to segment text corpora into smaller portions using various hierarchical clustering algorithms; these sub-corpora are then utilized in a Machine Translation pipeline. Section 5 sums up the discussion and concludes.

2 Background and Related Work

The discussions in Theoretical Linguistics on text types have mostly been using the term *sublanguages* for specific types of texts within one language. In the Machine Translation area, the term *language domain* is more frequent; however, there is no clear definition of what a domain is. Price (1990) talks about domains as the specificity of a sub-language. According to Kittredge and Lehrberger (1982) sublanguages vary in terms of: subject matter; lexical, syntactic and semantic properties; grammar rules; frequency of specific constructions (as idioms); text structure; and use of special symbols.

Karlgren (1993) pointed out the problematic sides of the mathematical readings of the terms ‘sub’ and ‘super’ in that sublanguages often will exhibit properties that are not present in the perceived *superlanguage*, the general language from which a sublanguage is divided. He instead proposed the term *register*, borrowed from Sociolinguistics. Registers are defined as varieties of language according to use, in contrast to varieties according to speaker or geographical location. This definition is therefore more process-oriented, to be understood as properties of the speaker directly.

In Bungum and Gambäck (2011), we gave an account of domain adaptation with a special emphasis on shared knowledge with cognitive sciences. There we observed how much work is relating to an original domain and a new domain, where a system suited for the former is adapted into fitting the latter. In the present work we are prototyping a system that will adapt a Machine Translation system simultaneously to many different domains and contexts. This means that there is a need to translate on the document level, that is, to adapt a general MT system according to the document properties.

Early attempts at adapting Statistical Machine Translation (SMT) models went into adapting the Language Model (LM), inspired by language model adaptation work in speech recognition (Iyer et al., 1997; Rosenfeld, 1996). Bellegarda (2004) gave a review of the work on Statistical Language Model Adaptation from the perspectives of Automatic Speech Recognition (ASR) at the beginning of the Big Data era, poignantly making the case for Domain Adaptation by pointing to both that Language Models are very brittle across domains (Rosenfeld, 2000) and that a small (2 million words) corpus can be better in perplexity terms than a large (140 million words) for a specific ASR task.

Building on Information Retrieval (IR) methods, Mahajan et al. (1999) used cosine similarity between document vectors to interpolate a general LM to a domain specific model computed on the retrieved documents, in what they called a Dynamic Language Model, where an auxiliary LM is built depending on the documents similar to the input document. Eck et al. (2004) also used IR techniques, retrieving the most similar documents and sentences from a large collection. They explored two aspects of adaptation: the best amount of documents to retrieve and the best unit of retrieval, noting that sentence level retrieval performed best in terms of perplexity reduction, but with a weak correlation between improvement of the LM and the overall SMT task. While this was not strictly speaking Domain Adaptation, as all the test data was of the same kind, the ideas and methods would still be applicable to different text types. IR-inspired strategies have also been applied by others, although with limited success: Wang et al. (2014) took an edit-distance approach based on normalized Levenshtein (1966) similarity, while Lu et al. (2007) added a log-linear interpolation of *sub-models*, i. e., models built on data from each of the domains.

Moore and Lewis (2010) proposed a method of selecting relevant sentences from an out-domain corpus to build an auxiliary Language Model. They experimented on the Europarl and Gigaword corpora, with the former defined as in-domain text. The method selected sentences from the Gigaword corpus by comparing the difference in Cross-Entropy of an LM built on the English-French part of Europarl to one based on a random selection of sentences from Gigaword. The Gigaword sentences were then split into eight equal portions, ranked after this difference, from which new LMs were built. The method was compared to other selection criteria, such as ranking the sentences only on the perplexity score from Europarl or scoring each Gigaword sentence on the log-likelihood of testing the Europarl corpus on unigram models built with and without this sentence, as well as a random selection of Gigaword sentences. The method based on Cross-Entropy difference had a lower Perplexity than all the other methods until all data was added, but more importantly, using a only small portion of the data gave a lower perplexity than adding the whole corpus. Axelrod et al. (2011) performed similar extraction of *pseudo-in-domain* sentences based on Cross-Entropy measures; pseudo because they are similar, but not identical to the in-domain data. These extracted corpora were used to train smaller domain-adapted translation models that performed better for the target context than a model built on the entire material, and improved even more in combination. Discarding as much as 99% of an original, general-purpose corpus, Axelrod et al. (2011) achieved better results with the pseudo-in-domain model.

There have been some notable efforts of injecting Domain Adaptation in a full Statistical Machine Translation pipeline. Carpuat et al. (2012) performed experiments on Phrase-Sense Disambiguation, a discriminative approach to MT where the correct Phrase is chosen according to a number of criteria at decode-time. Carpuat and Wu (2007) discuss how the method can be integrated into an SMT framework by means of dynamic phrase tables. However, the approach yielded no improvements in their experiments. The Moses SMT toolkit (Hoang et al., 2007) was used by Louis and Webber (2014) with cached language models to store domain specific information, and by Sennrich (2011) to build "dynamic" phrase tables, after an addition to a limited in-domain parallel corpus, and to combine these with Moses' log-linear decoding procedure. Going further, Sennrich et al. (2013) investigated the idea of using unsupervised methods to classify text, combining them with mixture models to perform Domain Adaptation.

3 Methods and Data

We have formulated a Domain Adaptation problem that requires two steps. First an outward-looking approach making use of additional, unlabelled training material, and second a way to include this dynamically in a Machine Translation pipeline. An unsupervised algorithm, the Self-Organizing Map (SOM) is employed to induce structure from an unstructured body of text and to create auxiliary Language Models for SMT decoding. The number of clusters to be created from the SOM is decided through bottom-up hierarchical agglomerative clustering. The method provides n separate clusters of text, from which standard n-gram Language Models are built to be used as auxiliary LMs in Domain Adaptation. In this way, the number of auxiliary text corpora (and later LMs) are determined by the clustering algorithm, enabling Domain Adaptation into the *available* domains, which is why an unsupervised method was chosen.

3.1 Data

Two data collections have been used in this work. The first, the SdeWac corpus (Faa and Eckart, 2013) consists of 10,830 unorganized German web documents, 9,056 of which were included here, comprising 8.1G of text. *Unstructured* text corpora with distinct features were extracted from SdeWac to create clusters for later use in Domain Adaptation. Each document in the collection was vectorized with Scikit-learn (Pedregosa et al., 2011), which includes several different vectorizers, on word and character level. A unigram TF-IDF (term frequency-inverse document frequency) vectorizer was mostly used in these experiments. Some tests were also run combining it with n-gram frequencies and on bigrams.

The second data collection was created from standard Statistical Machine Translation training and test corpora. A baseline SMT model was trained on Europarl version 7 (Koehn, 2005) and News Commentary (Tiedemann, 2012), containing news text and commentaries from the Project Syndicate. Two datasets were used for development (for tuning weights), the EMEA and Subtitles corpora, while a held-out Newstest corpus was used for testing (evaluation).

3.2 Self-Organizing Map as Clustering Device

Self-Organizing Maps are created with an unsupervised algorithm that effectively visualizes underlying structure of complex data in a 2D environment, in that respect a dimensionality reduction. Pioneering research into computer simulations of self-organizing systems was conducted at the Helsinki University of Technology (Kohonen, 1982) with Kohonen et al. (1996) employing it to self-organize usenet (newsgroup) data, Kohonen et al. (2000) to cluster patent data, and Lagus et al. (2004) the Encyclopedia Britannica.

The underlying idea has its basis in neuroscience and assumes that unordered inputs can be mapped to a “topological order” through a self-organzing process. The process starts with an array of randomly initialized nodes according to some topology, each node represented by a vector of the same dimension as the training data. During training, input samples are compared to the vectors, and the node whose vector is closest to the input is considered winner (Best-Matching Unit). The winner’s weights, as well as those of the neighboring nodes, are updated to be more similar to the input sample. The degree to which the weights are updated, both for the winning node and its neighborhood, is parameterized. The process is repeated for a (often pre-defined) number of iterations. As the iterations progress, the learning rate (the degree at which the nodes update) diminishes according to a parameterized function. The resulting Self-Organizing Map gives insights into the relations between the input samples: similar samples should attach to adjacent nodes, hence *self-organizing* into the same area.

The areas in the Self-Organizing Map are not delimited. In order to transform them into actual clusters, it is necessary to perform clustering on the map, represented as a grid of vectors. To cluster these vectors, a *metric* to determine similarity between clusters is chosen, alongside a *method* to select which data points the distances are measured between. According to these two facilities, the nodes are merged successively from the bottom up. Some clustering methods compute distances between nodes based on raw observations (node vectors), but more commonly a distance metrics, such as Euclidean or Chebychev distance, is used to construe a *Distance* Matrix containing the distances between nodes in the grid. Based on these distances, a *Linkage* Matrix is computed, that contains the binary links between nodes. This can be represented with a dendrogram.

Clustering Methods Hierarchical agglomerative clustering algorithms are algorithms that successively merge clusters. Whereas a top-down approach needs metrics to split clusters, a bottom-up approach needs criteria to merge clusters. The number of clusters that come out as a result from the whole clustering procedure is determined by choosing the height of the tree. At its minimum it is only one cluster, and the maximum is the number of nodes in the tree; in our experiments the number of nodes in the Self-Organizing Map.

The number of clusters can either be specified directly, or chosen according to some criteria. There is no analytical way to determine the best number of clusters (ultimately it depends on the task clustering for). Salvador and Chan (2004) present several methods to determine the number of clusters automatically, such as information theoretic methods that quantify the fit of cluster members to their centroids and optimize the number of clusters on this, or the elbow/knee method, that finds the point before merging two clusters that will add the most to the distance between clusters (i. e., the point where the marginal gain of adding another cluster is the lowest). Determining the second derivative of the points is one way to find the knee, and we have used this method in our experiments.

The clustering method, that is, the algorithm that measures the distance between two clusters, is a choice of what points to compare distances between. The methods that have been explored in this work are *single-linkage*, *average distance*, *complete linkage*, *weighted linkage*, *Ward clustering*, *centroid clustering*, and *median*. In single-linkage clustering, the distance between two clusters is determined as the distance between the two points in the clusters that are closest. On the other extreme, complete linkage uses the maximum distance between two points in the clusters, and the average method the average distance between the points in the two clusters. The weighted method takes into account the children of a cluster as well, using the average of the distance between the two clusters that formed cluster A and B as the distance between the two.

Evaluation Evaluating the unsupervised clustering process is two-fold; first the clustering can be evaluated *intrinsically* according to properties of the mathematical relation between the nodes, or *extrinsically* according to the performance gain or loss on the task for which the clustering was done. For extrinsic evaluation, we applied the corpora resulting from the above clustering method and created Language Models from them; models that were then used as auxiliary LMs in a Statistical Machine Translation system.

For intrinsic evaluation we have used the Silhouette Coefficient, which is a measure of the clusters' separation and how compact they are. Following Han (2005, pp 489-490), the Silhouette is calculated by first determining $a(o)$, the average distance between an object o $[o \in C_i (1 \leq i \leq k)]$ and all other objects in the cluster C to which o belongs: $a(o) = \sum_{o' \in C_i, o \neq o'} \text{dist}(o, o')/(|C_i| - 1)$, and then $b(o)$, the minimum average distance from the object to all other clusters to which it does *not* belong: $b(o) = \min_{C_j : 1 \leq j \leq k; j \neq i} \sum_{o' \in C_j} \text{dist}(o, o')/|C_j|$. Then the Silhouette Coefficient is determined by

$$s(o) = \frac{b(o) - a(o)}{max\{a(o), b(o)\}} \tag{1}$$

The value of the Silhouette Coefficient will range between $[0, 1]$. A higher number indicates a better clustering. The term $a(o)$ is an indication of *compactness*. A lower value indicates a more compact cluster, i. e., low distance between the members inside a cluster, whereas $b(o)$ describes the separation from other clusters. The larger it is, the more separated the cluster is from the others. This means that when the Silhouette Coefficient reaches 1, the clusters are compact [low $a(o)$] and far away from the other clusters [high $b(o)$]. It would also be possible to optimize the number of clusters based on this score.

4 Domain Adapted Machine Translation

The strategy chosen for domain adapted Machine Translation is to first segment large corpora with a Self-Organizing Map and then utilize language models built on the basis of these corpus segments in a Statistical Machine Translation system. The first phase is conducted off-line, whereas the employment of the language models is done on-line, while decoding an SMT model given input sentences.

When a document is input for translation, it is matched against the n created Language Models, ranked after perplexity. The Moses SMT system (Koehn et al., 2007) allows for the use of user-defined features in its log-linear model. The LM with the lowest perplexity is selected by a new Moses feature providing 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. The Language Models where built with the KENLM toolkit (Heafield, 2011), which was also used to read the LMs when decoding the SMT models.

4.1 Clustering Experiments

A quadratic layout of nodes in a Self-Organizing Map was used in the experiments, and for each sample the comparisons to all the different vectors in the node were done in parallel, as was the updating of the nodes in the next stage. Each SOM was configured with a separate configuration file, where the number of iterations, the initial size of the neighborhood (given as a radius in Euclidean space), and the initial learning rate were specified. In this file, details about the vectorization of the document collections could also be entered.

After a pre-defined maximum number of iterations was reached, 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, as described above.

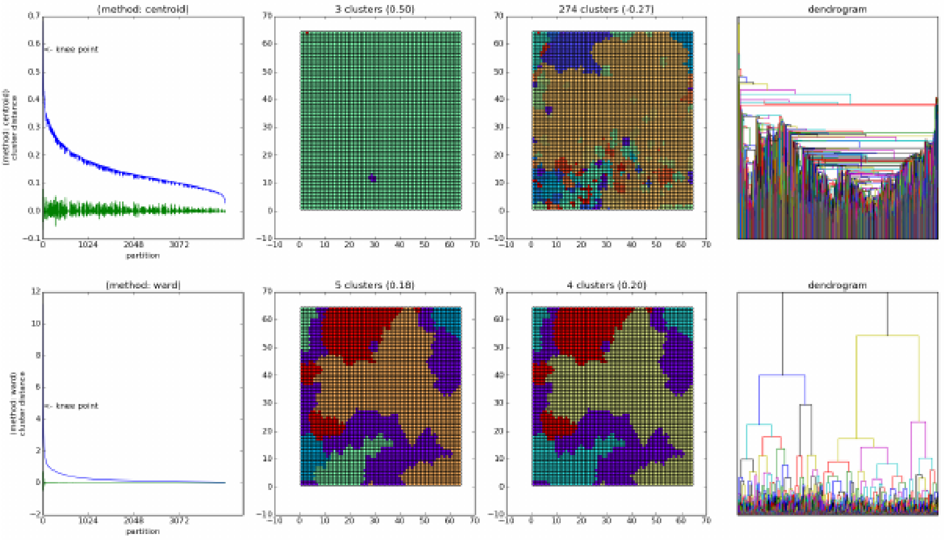

Fig. 1: Hierarchical clustering (centroid and Ward) for a 64x64 SOM.

In Figure 1, the detailed clustering of the SdeWac corpus using the first and second knee-points is shown. The knee-points are in the first column, the resulting clusterings in the 2nd and 3rd columns, and the corresponding dendrogram in the 4th. The two methods *centroid* and *Ward* exemplified here illustrate the differences in results between methods. Ward's method generally resulted in more evenly sized clusters and was used as a basis for the further experiments. The color codings come from Principal Component Analysis (PCA) decomposition of the node vectors into four dimensions. These plots do not, however, show the size of the clusters as some nodes will contain more samples than others. This can be illustrated with heat maps, but those are left out here for space reasons.

The creation of Self-Organizing Maps is subject to many variables, such as grid size, vectorization methods, and distance metrics. We conducted several clustering experiments on the SdeWac corpus with different SOM sizes, summarized in Table 1. The table shows the Silhouette Coefficients for the partitionings, with the number of clusters in each in parentheses. The two sets of experiments come from two different vectorizations of the SdeWac data. In the first experiments, the TF-IDF for the n-grams up to 7 were computed with a fixed vocabulary from the entire corpus, but on a subset of 1,000 documents (dimensionality 157,053). In the second set of experiments, the vectorization process had the same cut-off points, but the n-grams were computed only from that part of the corpus (dimensionality 53,016).

We then went on to use one of these experiments for Domain Adaptation, presented in the next section.

Table 1: Silhouette scores for corpora clustered with SOM.

Grid size	8x8	16x16	32x32	64x64
1st Part. (#)	0.33 (2)	0.46 (2)	0.35 (2)	0.31 (2)
2nd Part. (#)	0.23 (4)	0.31 (10)	0.31 (4)	0.29 (4)
1st Part. (#)	0.39 (2)	0.41 (7)	0.37 (3)	0.33 (2)
2nd Part. (#)	0.30 (3)	0.34 (11)	0.32 (8)	0.32 (4)

4.2 Using Self-Organized Clusters for Domain Adaptation

A Statistical Machine Translation model consists of a Translation Model and a Language Model, that are decoded to provide the optimal Target Language string given the input. The term *auxiliary Language Model* denotes an additional LM used to aid this decoding. We have shown above how the text collections are divided into sub-corpora, through a Self-Organizing Map and an Agglomerative Clustering procedure. A selection method among the sub-corpora is necessary in order to test our hypothesis that building an Language Model over the most relevant sub-corpus will give nearly as good results as using the entire text.

With training data available in the target domain, the perplexity of the Target Language version of the development set can be used to rank the auxiliary LMs. In the absence of such data, a two-pass solution is possible, i. e., that a document is translated with the baseline Machine Translation first, and the perplexity of this translation is used to rank the LMs. This is the method we have applied in this work. Other possible selection criteria include matching a vectorization of the input document against the Self-Organizing Map used as a basis for clustering, using the cluster to whose SOM node it matched at closest. Such matching can be done on sentence, document or collection level.

An extra Language Model can be included in a Machine Translation system in a number of ways; concatenating the text corpus with the original LM, using an additional LM in Moses' configuration file or interpolating Language Models directly. When using such features, their individual weights can be optimized using the Minimum Error Rate Training (Bertoldi et al., 2009) training procedure. When using prepared datasets for Domain Adaptation research, it is reasonable to treat the entire sub-part of the corpus belonging to the same text type as one, a given premise in how they are developed. Then all sentences can be tested in the SMT system against the auxiliary LM found most relevant. However, we also developed a separate Moses feature that uses a specific LM for each input sentence, depending on which of them was closest according the selection criterion mentioned above. Concretely, this information was looked up in a file that was compiled from labeling individual sentences with the most relevant sub-corpus (chosen as described in the previous paragraph).

LM	EMEA	Subs	News
1	**25,269**	281	893
2	41,386	374	1030
3	25,494	**261**	**744**
4	30,320	556	1030

(a) including

LM	EMEA	Subs	News
1	12,501	182	470
2	13,764	242	510
3	12,566	**167**	**406**
4	**11,260**	331	494

(b) excluding

Table 2: Perplexity scores including and excluding OOV words. Low scores in bold.

LM	BLEU	Meteor
LM1	*19.28*	***0.3492***
LM2	17.97	0.3358
LM3	18.55	0.3418
LM4	18.43	0.3420
No Aux.	16.82	0.3347
Total	**19.29**	0.3479

(a) EMEA

LM	BLEU	Meteor
LM1	12.85	0.3332
LM2	12.47	0.3366
LM3	*13.05*	***0.3367***
LM4	12.14	0.3304
No Aux.	11.29	0.3236
Total	**13.09**	0.3366

(b) News

LM	BLEU	Meteor
LM1	**10.53**	0.**2704**
LM2	9.93	0.2646
LM3	*10.01*	*0.2670*
LM4	8.88	0.2551
No Aux.	8.48	0.2563
Total	10.47	0.2668

(c) Subtitles

Table 3: Domain Adaptation results on English-German text across different LM configurations. Highest score in bold, LM with lowest perplexity in italics.

4.3 Domain Adaptation Experiments

As a baseline system, we used the data presented in Section 3. An SMT model was created from parliamentary text, and three domain specific corpora were used for testing: medical text (*EMEA*), TV subtitles text, and a news corpus. The auxiliary LMs came from building a SOM trained on the SdeWac corpus that after Hierarchical Agglomerative Clustering with the Ward method resulted in four corpora, from which Language Models were built. In order to test which LM was the most relevant for each of the input documents, perplexity was measured on a) the translation of each test corpus provided by the parallel corpora and b) the translation of the test set provided by the generic SMT model.

The Ward clustering, that is, the number of clusters from the dendrogram, from the first partitioning (knee-point) was used in further experiments into Domain Adaptation. This meant three clusters for this data, and accordingly three text corpora and Language Models. The next step was to measure the perplexity of the three *in-domain* corpora, to determine which auxiliary LM was closest. The results are presented in Table 2. The *EMEA* corpus selected LM1, whereas the *Subs* and the *News* corpora selected LM3 according to this criterion.

In Table 3 the results from using these Language Models in the full Machine Translation pipeline are shown, using BLEU and Meteor evaluation. For the EMEA and News corpora, the LM with the lowest perplexity also gave the highest gain in MT performance of the auxiliary LMs, for the Subtitles corpus it came second. The results are also compared to not using any auxiliary LM and using all the text to build one large Language Model.

5 Discussion and Conclusion

The work presented here introduces a solution to the Domain Adaptation problem that looks for external sources to increase the available training data rather than exploiting some limited source of *in-domain* data in refined ways.

The purpose of the Language Model in a Machine Translation pipeline is to provide *fluency*, by selecting translation candidates that are likely members of the target language. The famous Firthian truism *"You shall know a word by the company it keeps"* speaks to understanding a word from its context. By mapping input documents to close-matching Language Models, we try to provide this context for arbitrary input documents.

However, it is not always obvious how to distinguish text domains from each other, where to draw the line between them and according to what dimensions (such as writing style, topic, author or target age groups) to separate them. An unsupervised approach to segmentation of a vast data source is a way of enabling a Machine Translation system to respond to various input domains also along such dimensions. The proposed Domain Adaptation method includes a step to establish which of the auxiliary Language Models is the most relevant for the given input document, and thereby enabling context adapted simultaneous translation of documents of different types.

The strategy builds on a Self-Organizing Map (SOM) approach to finding relations within unorganized data, a strategy which has also been successfully utilized in other application areas. Results indicate that it is possible to attain improvements equal to — or even better than — those stemming from adding the whole available text collection by using the most relevant part, also by unsupervised methods.

More research is necessary to establish the feasibility of the Self-Organizing Map to extracting relations useful for extra training material for Statistical Machine Translation models, notably into self-organizing parallel corpora.

References

A. Axelrod, X. He, and J. 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.

J. R. Bellegarda. 2004. Statistical language model adaptation: review and perspectives. *Speech Communication*, 42:93–108.

N. Bertoldi, B. Haddow, and J.-B. Fouet. 2009. Improved minimum error rate training in moses. *Prague Bull. Math. Linguistics*, 91:7–16.

P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

L. Bungum and B. 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.

M. Carpuat and D. Wu. 2007. How phrase sense disambiguation outperforms word sense disambiguation for statistical machine translation. In *Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation*, pages 43–52, September.

M. Carpuat, H. D. III, A. Fraser, C. Quirk, F. Braune, A. Clifton, A. Irvine, J. Jagarlamudi, J. Morgan, M. Razmara, A. Tamchyna, K. Henry, and R. Rudinger. 2012. Domain adaptation in machine translation: Final report. In *2012 Johns Hopkins Summer Workshop Final Report*. Johns Hopkins University.

M. Eck, S. Vogel, and A. Waibel. 2004. Language model adaptation for statistical machine translation based on information retrieval. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 327–330, Lisbon, Portugal, May. ELRA.

G. Faa and K. Eckart. 2013. Sdewac – a corpus of parsable sentences from the web. In I. Gurevych, C. Biemann, and T. Zesch, editors, *Language Processing and Knowledge in the Web*, volume 8105 of *Lecture Notes in Computer Science*, pages 61–68. Springer Berlin Heidelberg.

J. Han. 2005. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

K. Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, July. ACL.

H. Hoang, A. Birch, C. Callison-burch, R. Zens, R. Aachen, A. Constantin, M. Federico, N. Bertoldi, C. Dyer, B. Cowan, W. Shen, C. Moran, and O. Bojar. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

R. Iyer, M. Ostendorf, and H. Gish. 1997. Using out-of-domain data to improve in-domain language models. In *Signal Processing Letters*, pages 221–223.

IEEE, August.

J. Jiang. 2008. *Domain Adaptation in Natural Language Processing.* University of Illinois at Urbana-Champaign.

J. Karlgren. 1993. Sublanguages and registers — a note on terminology. *Interacting with Computers*, 5(3):348–350, September.

M. Kay. 1980. The proper place of men and machines in language translation. Technical Report CSL-80-11, Xerox Palo Alto Research Center, Palo Alto, California.

R. Kittredge and J. Lehrberger, editors. 1982. *Sublanguage: studies of language in restricted semantic domains.* W. de Gruyter, Berlin; New York.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. 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.

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

T. Kohonen, S. Kaski, K. Lagus, and T. 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*, Lecture Notes in Computer Science, vol. 1112, pages 269–274. Springer, Berlin, July.

T. Kohonen, S. Kaski, K. Lagus, J. Salojrvi, V. Paatero, and A. 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.

T. Kohonen. 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69.

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

V. I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, February.

A. Louis and B. Webber. 2014. Structured and unstructured cache models for smt domain adaptation. In I. Shuly Wintner, University of Haifa, G. Stefan Riezler, Heidelberg University, and U. 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*, Gothenburg, Sweden, April. Association for Computational Linguistics.

Y. Lu, J. Huang, and Q. Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 343–350, Prague, Czech Republic, June. Association for Computational

Linguistics.

M. Mahajan, D. Beeferman, and X. D. Huang. 1999. Improved topic-dependent language modeling using information retrieval techniques. In *in ICASSP*.

R. C. Moore and W. Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, volume Short papers, pages 220–224, Uppsala, Sweden, 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 Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(1):2825–2830.

B. Plank. 2011. *Domain Adaptation for Parsing*. Ph.d. thesis, University of Groningen.

P. J. Price. 1990. Evaluation of spoken language systems: The atis domain. In *Proceedings of the Workshop on Speech and Natural Language*, HLT '90, pages 91–95, Stroudsburg, PA, USA. Association for Computational Linguistics.

R. Rosenfeld. 1996. A maximum entropy approach to adaptive statistical language modeling. *Computer, Speech and Language*, 10:187–228.

R. Rosenfeld. 2000. Two decades of statistical language modeling: Where do we go from here. In *Proceedings of the IEEE*, page 2000.

S. Salvador and P. Chan. 2004. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *ICTAI*, pages 576–584. IEEE Computer Society.

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

R. Sennrich. 2011. Combining multi-engine machine translation and online learning through dynamic phrase tables. In *EAMT-2011: the 15th Annual Conference of the European Association for Machine Translation*, Leuven, Belgium, May. European Association for Machine Translation.

J. Tiedemann. 2012. Parallel data, tools and interfaces in opus. In N. C. C. Chair), K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

L. Wang, D. F. Wong, L. S. Chao, Y. Lu, and J. Xing. 2014. A Systematic Comparison of Data Selection Criteria for SMT Domain Adaptation. *The Scientific World Jornal*, 2014(1).