Achieving Domain Specificity in SMT without Overt Siloing

William D. Lewis, Chris Wendt, David Bullock

Microsoft Research
Redmond, WA 98052, USA
wilewis@microsoft.com, christw@microsoft.com, a-davbul@microsoft.com

Abstract

We examine pooling data as a method for improving Statistical Machine Translation (SMT) quality for narrowly defined domains, such as data for a particular company or public entity. By pooling all available data, building large SMT engines, and using domain-specific target language models, we see boosts in quality, and can achieve the generalizability and resiliency of a larger SMT but with the precision of a domain-specific engine.

1. Introduction

Statistical Machine Translation (SMT) is now seen as commercially viable, and a burgeoning market of "enterprise MT" based on Statistical MT engines is evolving. MT is especially useful for content that changes or is updated frequently (such as Websites), where large data stores (such as knowledge bases) need to be translated on-the-fly, where translation engines can be trained and retrained as needed, where some sacrifices of quality are permissible, and where standard approaches to localization, namely manual translation, are fiscally and logistically impossible. In the enterprise space, there is often a high degree of homogeneity in the data and SMT often performs quite well. As is typical with SMT, however, significant amounts of parallel training data may still be required. For many, large amounts of parallel training data may be difficult to come by. Increasing the amount of training data through diversification (e.g., using out-of-domain, heterogeneous supplies of data), however, can lead to drops in quality, as measured by both BLEU and human eval. The quality barrier is then limited by the amount of in-domain parallel training data, a problem when it is in short-supply.

We have been pursuing methods of developing domain specific SMT by tapping large pools of heterogeneous data without sacrificing quality within specific domains. Such research is not novel: adapting SMT to specific domains has been pursued in a number of venues, most notably in the Workshops on Statistical Machine Translation (WMT) shared tasks focused on domain adaptation.[1] Typically, however, *domain* is interpreted rather broadly, e.g., Europarl, Newswire, etc. Here, we interpret *domain* very narrowly, e.g., the document supply for a particular entity, such as a commerical enterprise, public institution, etc. (what might be best labeled as *micro-domains*). We study here the lower bounds of in-domain data and its impact on quality, and to what extent pooling out-of-domain sources can be used to achieve quality boosts and how great these quality boosts can be.

[1] http://www.statmt.org/wmt08/

1.1. Language Models and Domain Specificity

It is generally accepted as a given that the more training data one has, the better the quality of the SMT built on that data. However, if one's focus is on building a domain specific engine, pooling together all available data, especially a significant portion of data that is out of the desired domain, can lead to reductions in quality, since the out-of-domain training data will overwhelm the in-domain (Koehn and Schroeder, 2007). Unfortunately, the drawback of domain specific SMT, that is, where only in-domain data is used, is its failure to capture generalizations relevant to the target language, which can lead to poor translation quality (Thurmair, 2004). What is desirable in a domain specific engine is to capture the generalizations of an engine trained on a large and sufficient supply of parallel data, yet not lose the crucial domain orientation of a an SMT, namely, one that preserves domain-specific word and phrase meanings, domain-specific phrasing, etc. To achieve this, we can train on all available data, yet split language model training data (minimally) into in-domain and out-of-domain sets, generating separate LMs for each (as explored in (Koehn and Schroeder, 2007)). If we use domain specific development to produce lambdas that favor the domain-specific LM over the out-of-domain one (effectively the out-of-domain becomes a backoff model), we can achieve domain-specificity without sacrificing some degree of generalizability.

Obviously, our language models must contain hypotheses that are possible output from our translation model. It is therefore essential that our parallel training data and the data we use for training our language models be somewhat in concord. We can achieve this by using the same data for both: we train the translation model on both sides of a parallel corpus and train language models on the target side of the same data (of course, splitting the

language models appropriately). As long as we do not aggressively prune any model (or prune them in compatible ways), we can ensure that any hypothesis produced by our translation model can be computed against our language models.

So, to achieve domain specificity, we need to (1) assemble as large a corpus as possible of domain specific data and train a translation model on that data, and (2) assemble a large corpus of data irrespective of source, and train a secondary language model on that data. Our only requirement will be to have an ample supply of in-domain parallel and monolingual data. [2]

As noted in the Introduction, the term *domain* is usually interpreted broadly, representing broad categories, such as government, newswire, entertainment, travel, sports, etc. We interpret domain very narrowly in this paper, where a domain represents data for a specific firm. Generally, the more narrow the domain, that is, the more reduced the set of possible hypotheses that can be represented in an LM and that can be output by a translation model, the less training data for both that will be required. However, given *enough* in-domain parallel data we may be able to forego (2) in favor of an SMT built just on (1). (See Section 6. for a preliminary discussion and analysis.)

2. Microsoft SMT Environment and Related Resources

Microsoft's Machine Translation engine (Menezes and Quirk, 2005; Quirk et al., 2005),[3] as with any SMT engine, relies heavily on parallel data to build the relevant models (see Figure 1 for the design of the Microsoft Translation engine). Further, monolingual data is used to build Language Models (LMs), which contain a probabilistic space for testing translation hypotheses. We use Minimum Error Rate Training (MERT) (Och, 2003) against held-out development data to train the feature weights for our LMs, using random restarts as discussed in (Moore and Quirk, 2008). The target side of training data is automatically used for such LMs, but additional monolingual data can be added to increase LM size and utility.

Since Microsoft has been localizing its products into a large number of languages for many years, we have developed a large data store of multilingual localized content. If we train an engine on the localized content for some given language pair, say English-German, the resulting engine performs quite well on similar English input. An engine trained on broader coverage, more heterogeneous data, however, tends to do less well. As an example, note the differences between our General Domain and Microsoft engines shown in Figure 2.[4] The Microsoft engine was trained on homogeneous Microsoft Localization data, and the General Domain engine was trained on a diverse set of data from many sources, e.g., Web, newswire, etc., but also a large amount of our own localization data. The results for General Domain system clearly demonstrate the disadvantage of training a system on pooled data; out-of-domain data clearly affects quality on in-domain content.

3. TAUS Data and Test cases

The TAUS[5] Data Association (TDA) recently launched the TDA language data exchange portal. The portal allows members to freely exchange translation memories (TMs) and vocabularies, and at launch consisted of 500M words in 70 languages. Although the TDA is certainly useful for pooling TMs for traditional localization efforts, the pool of data can be used for training SMT engines as well. The difficulty lies in how best to use the data, especially if one wishes to localize to a particular data provider. The data providers who have English-German data and the corresponding number of segments is shown in Figure 3.

From our experience, large quantities of data are required to train an engine, often hundreds of thousands to millions of segments.[6] One can see a dramatic demonstration of the data requirement by removing data from a system, and seeing the resulting effects on the automated evaluation metric BLEU (Papineni et al., 2002). For instance, if we trim the data used to train the Microsoft system from 7.6M segments to 500K (randomly sampled from the 7.6M), BLEU drops precipitously from 52.39 to 37.68. Such a drop in BLEU would result in a significant drop in output quality if such a system were to be used.

Suppose that some TDA member, say Sybase, Dell, or Adobe, wished to train a highly specific SMT on their data. Following the traditional model, that is, training a specific engine strictly on in-domain text (i.e., producing a *silo*), there might not be a sufficient supply of

[2] In (Moore and Lewis, Under Review) we discuss methods for generating in-domain language model training data from out-of-domain sources using models built over in-domain data. Obviously, if successful, such work can increase the body of data that resembles in-domain data, which could then be used to improve the quality of domain-specific translation systems.

[3] A publicly available version is available at: http://microsofttranslator.com.

[4] Each eval set consists of 5,000 segments, one reference. The Microsoft eval set consists of data held out from our localization data store. The General Domain set consists of independently collected segments representing frequent translations.

[5] Translation Automation User Society, http://www.translationautomation.com

[6] Segments generally equate to sentences, however, some training and test data can consist of sentence fragments, named entities, etc. We therefore avoid the use of the frequently used term *sentence* because of this variability.

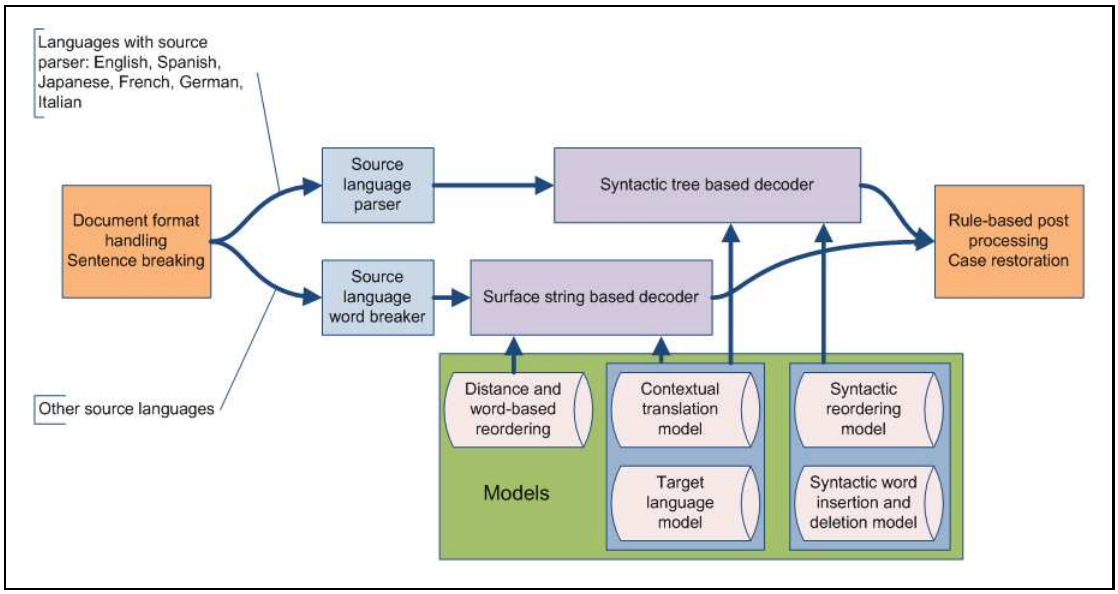

Figure 1: Microsoft Translator

		Test Set	
System Size	**System Description**	**General**	**Microsoft**
4.4M	General Domain	25.19	40.61
7.6M	Microsoft	21.95	52.39

Figure 2: German Cross-Wise Comparison

Provider	# Segments
EMC	414791
Intel	128209
Dell	314496
eBay, Inc.	59967
Avocent	93498
EMC	124065
McAfee	497938
Sybase iAnywhere	216315
ABBYY	28063
Adobe	232914
Sun Microsystems	51644
PTC	178341
Intel	11566
SDL	44029
Microsoft	6172394

Figure 3: TDS Data Providers, and Amounts of Data (for German)

provider-specific English-German data to ensure producing an engine of reasonable quality. The solution lies in how to best use the supply of available training data, and "tune" whatever engine results to the input for a particular provider.

4. Building Domain Specific MT engines

In the enterprise space, the availability of open-source tools, freely available APIs, and readily available training data in a number of languages, it is now possible to create SMT engines with minimal initial investment. The true costs lie in providing an adequate supply of in-domain data over which models can be built.[7] A firm that wishes to localize their content into a dozen different languages would face the daunting challenge of first localizing some of the content before being able to train an SMT of their own.[8] However, unless the firm is able to compile a ample set of parallel training data—generally,

[7]Rule based MT (e.g., (Dugast et al., 2007)) does not have the data limitations of Statistical MT. However, there are costs incurred in developing the rules (work will vary depending on the divergence of the new rules from those in an existing engine) and in tailoring the rules for a specific domain. This paper will not review rule-based approaches, nor their benefits or disadvantages as compared to SMT.

[8]Granted, they could use a third party engine and post-edit the content in order to reduce the initial investment.

on the order of hundreds of thousands of words—the resulting engine may be of limited utility. Granted, over time, as more data is translated and added to the pool of training data, quality will improve.

As noted in the previous section, we can achieve in-domain quality boosts by "pooling" in-domain and out-of-domain data. The problem is finding sufficient in-domain and out-of-domain sources. The TDA may provide an ample supply of data that can be used.

5. Building a Provider Specific SMT Engine

We ran a series of experiments using various combinations of data from the TDA data pool, our own localization data, and a portion of our General Domain training data. We chose one TDA data provider, Sybase, for our first set of experiments. The focus was to develop a translation engine of reasonable quality for that provider using whatever combination of resources worked best. Our first baseline (1) is a General Domain engine built on 4.4M segments. Our second (2a) is the Microsoft engine, trained exclusively on 7.6M segments of Microsoft localization data. The comparison system (2b) was the same as (2a), except all available Sybase data was also used, excepting 7,000 segments: 5,000 for eval, and 2,000 for development data (for subsequent experiments). (We also built a Sybase-only system, but review of that system is saved for Section 6.).

The results of the three systems are shown in Figure 4. The Microsoft System (2a) performed much better than the General Domain system (1) on the Sybase eval set, suggesting similarity between the Microsoft and Sybase data sets (both consist of localization data). However, the Sybase data is still far more "in-domain" than the Microsoft data, since adding a small amount of Sybase data (210K segments) to a large Microsoft data pool (7.6M segments) caused a jump of over 1/2 BLEU point (41.55 to 42.07). Still, the 42.07 we see on the Sybase eval set for this system is much less than the 52.07 we see on the Microsoft eval data.

Given 2a's performance on the Sybase eval set, we suspected that additional TAUS data would help with boosting quality on that eval set (assuming a comparable degree of similarity with the Microsoft localization data). We built an additional system which pooled all TAUS data with the Microsoft data. To ensure broad vocabulary coverage, we also added the General Domain data, resulting in a system built over 11.1M segments (3a). This system performed even better on the Sybase eval set, increasing BLEU from 42.07 to 48.83, a nice jump in quality. For (3a), instead of building just one LM, we built two: one over the Microsoft and TAUS data (including Sybase), and one over everything else.

Source	# Segments
General	4.3M
Microsoft	3.2M
TAUS	1.4M
Dell	172K

Figure 6: Japanese Data Used in the Dell Experiments

Increasing the size of the training data, and splitting the LMs, improved quality on the Sybase eval set, with the additional TAUS data doing the additional work. Since much of the hypothesis space in the LM for (2b) resulted from Microsoft data, increasing the relevance of the Sybase hypotheses may help even further. For our next experiment (3b), we separated the data used to build the LMs in (3a) into three parts: Sybase only data (210K segments of Sybase target language data), Microsoft and TAUS (excluding Sybase), and everything else. Doing this resulted in an even larger jump in BLEU on the Sybase eval set, increasing it from 48.83 to 50.85. The results are shown in Figure 4. Given the very small size of the training data for the Sybase LM, the resulting boost in BLEU is remarkable. Note that the addition of the Sybase LM in (3b) caused a drop in the Microsoft eval set.

To ensure that our results were not artifacts of the Sybase data, language specificity (e.g., typological similarities between English and German, *a la* (Fox, 2002)), etc., we repeated experiments on data for another TDA provider, Dell, and for another language pair, English to Japanese. The results for these experiments are shown in Figure 5. The composition of the data for these systems was similar to those used for Sybase, as shown in Figure 4. These results confirm what we saw with the Sybase English-German experiments.

6. Homogeneity of Data and MT Quality

In the previous sections, we argued that "more is better", an argument that is generally true for SMT, that is, the more data one has the better the resulting SMT engine. We also propose a corollary in the case of domains, where "more" needs to be tempered by domain specific language models. What we did not show in Section 5. is how well we might fare on a domain if we did not pool data, that is, if we built systems on just the data for specific TAUS providers. Figure 6. shows the BLEU scores of experiments using pooled data similar to Systems (3a) (as shown in Figures 4 & 5., i.e., the builds that include all general domain Web data, MS Localization data, all TAUS data, etc.). Note that in most cases, BLEU scores increased when data was pooled (the results for Sybase and Dell are repeated here). The three exceptions are Adobe German, Adobe Chinese, and ZZZ

			Test Set		
System	Size	System Description	General	Microsoft	Sybase
1	4.4M	General Domain	25.19	40.61	34.85
2a	7.6M	Microsoft	21.95	52.39	41.55
2b	7.8M	Microsoft with Sybase	22.83	52.07	42.07
3a	11.1M	General and Microsoft and TAUS	23.86	52.72	48.83
3b	11.1M	System 3a with Sybase lambda	19.44	37.27	50.85

Figure 4: Sybase Experiments (English-German)

		Test Set		
System Size	System Description	General	Microsoft	Dell
1	General domain	17.99	37.88	26.72
2a	Microsoft	17.28	41.32	32.64
2b	Microsoft with Dell	14.76	30.87	39.49
3a	General and Microsoft and TAUS	17.33	42.30	39.89
3b	System 3a with Dell lambda	14.85	32.21	42.43

Figure 5: Dell Experiments (English-Japanese)

Chinese[9] which actually show a significant reduction in quality when the data is pooled (despite domain specific LMs). Also note that Sybase is virtually unaffected. We hypothesize that data providers whose data is less "diverse" gain less from data pooling than others whose data is more diverse. In a less diverse data set, the individual segments tend to be similar to one another, e.g., very similar grammatical structures, reduced vocabulary, increased instances of duplicates or near-duplicates, etc. In other words, the less diverse a set of training data, the less data will be required to build a system of reasonable quality, assuming, of course, that held-out test data is a measure of expected input. The difficulty lie in how best to measure diversity. We are currently examining measures of data diversity, such as vocabulary saturation, word edit distance, and perplexity, and how these correlate with measures of SMT quality. Results are forthcoming.

7. Conclusion

SMT typically requires a large amount of data to produce engines of high quality. The data barrier has traditionally been a limit to developing quality engines in scenarios where there is little "in-domain" training data, in effect, where siloing is not possible. Increasing supplies of data through diversity has the consequence of lowering the quality of the resulting engines when applied to domain specific text. However, we have demonstrated that it is possible to see benefits from a large supply of out-of-domain data, yet not sacrifice the utility of in-domain text, even if the latter supply is very small. The use of domain-specific LMs with engines trained on diverse stores of data offers promise for training in-domain SMT engines without sacrificing quality within these domains. We feel that the work we describe here can be applied not only to "Enterprise MT", but also to scenarios with more broadly defined domains.

But, as noted in Section 6., it is important to recognize that data pooling does not always work, even in highly restricted micro-domains. We show, for instance, that it is possible to achieve high-quality domain-specific engines with little training data, and in some cases, such systems perform better than those where the data has been pooled. We are currently analyzing why this may be the case using various measures of diversity. It is important to note, however, that such systems will be much more "brittle", that is, less resilient to new vocabulary or input data that is divergent from the training data. Train-

Provider/Language	BLEU 3a	BLEU Provider Only	# Segments
Adobe/CHS	28.44	33.13	80002
Adobe/DEU	30.97	36.38	165203
Adobe/PLK	33.74	32.26	129084
Dell/JPN	42.43	40.85	172017
eBay/ESN	51.94	45.50	45535
Sybase/DEU	50.85	50.23	160394
ZZZ/CHS	32.72	34.81	173892
ZZZ/ESN	54.26	52.12	790181

Figure 7: Japanese Data Used in the Dell Experiments

[9]ZZZ is an anonymization of a company name for a provider whose data is not being provided through TAUS, and whose name we could not reveal.

ing on large amounts of data provides two crucial advantages: a larger vocabulary, and more contexts per term, something not as easily achieved with small, impoverished training data sets.

8. References

Loïc Dugast, Jean Senellart, and Philipp Koehn. 2007. Statistical post-editing on systran's rule-based translation system. In *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague.

Heidi Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proceedings of EMNLP 2002*, Philadelphia, Pennsylvania.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague.

Arul Menezes and Chris Quirk. 2005. Microsoft Research treelet translation system: IWSLT evaluation. In *Proceedings of IWSLT 2005*, Pittsburgh, PA.

Robert Moore and William Lewis. Under Review. Intelligent selection of language model training data.

Robert Moore and Chris Quirk. 2008. Random restarts in minimum error rate training for statistical machine translation. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, Manchester.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan.

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 ACL*, Philadelphia, PA.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of the 43rd ACL*, Ann Arbor, MI.

Gregor Thurmair. 2004. Comparing rule-based and statistical mt output. In *Workshop on the Amazing Utility of Parallel and Comparable Corpora, LREC*, Lisbon.