Domain Adaptation for Machine Translation by Mining Unseen Words

Hal Daumé III
University of Maryland
Collge Park, USA
hal@umiacs.umd.edu

Jagadeesh Jagarlamudi
University of Maryland
College Park, USA
jags@umiacs.umd.edu

Abstract

We show that unseen words account for a large part of the translation error when moving to new domains. Using an extension of a recent approach to mining translations from comparable corpora (Haghighi et al., 2008), we are able to find translations for otherwise OOV terms. We show several approaches to integrating such translations into a phrase-based translation system, yielding consistent improvements in translations quality (between 0.5 and 1.5 Bleu points) on four domains and two language pairs.

1 Introduction

Large amounts of data are currently available to train statistical machine translation systems. Unfortunately, these training data are often qualitatively different from the *target* task of the translation system. In this paper, we consider one specific aspect of domain divergence (Jiang, 2008; Blitzer and Daumé III, 2010): the out-of-vocabulary problem. By considering four different target domains (news, medical, movie subtitles, technical documentation) in two source languages (German, French), we: (1) Ascertain the degree to which domain divergence causes increases in unseen words, and the degree to which this degrades translation performance. (For instance, if all unknown words are names, then copying them verbatim may be sufficient.) (2) Extend known methods for mining dictionaries from comparable corpora to the domain adaptation setting, by "bootstrapping" them based on known translations from the source domain. (3) Develop methods for integrating these mined dictionaries into a phrase-based translation system (Koehn et al., 2007).

As we shall see, for most target domains, out of vocabulary terms are the source of approximately half of the additional errors made. The only exception is the news domain, which is sufficiently similar to parliament proceedings (Europarl) that there are essentially no new, frequent words in news. By mining a dictionary and naively incorporating it into a translation system, one can only do slightly better than baseline. However, with a more clever integration, we can close about half of the gap between baseline (unadapted) performance and an oracle experiment. In most cases this amounts to an improvement of about 1.5 Bleu points (Papineni et al., 2002) and 1.5 Meteor points (Banerjee and Lavie, 2005).

The specific setting we consider is the one in which we have plentiful parallel ("labeled") data in a source domain (eg., parliament) and plentiful comparable ("unlabeled") data in a target domain (eg., medical). We can use the unlabeled data in the target domain to build a good language model. Finally, we assume access to a very small amount of parallel ("labeled") target data, but only enough to evaluate on, or run weight tuning (Och, 2003). All knowledge about unseen words must come from the comparable data.

2 Background and Challenges

Domain adaptation is a well-studied field, both in the NLP community as well as the machine learning and statistics communities. Unlike in machine learning, in the case of translation, it is not enough to simply

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 407–412,
Portland, Oregon, June 19-24, 2011.

adjust the weights of a learned translation model to do well on a new domain. As expected, we shall see that unseen words pose a major challenge for adapting translation systems to distant domains. No machine learning approach to adaptation could hope to attenuate this problem.

There have been a few attempts to measure or perform domain adaptation in machine translation. One of the first approaches essentially performs test-set relativization (choosing training samples that look most like the test data) to improve translation performance, but applies the approach only to very small data sets (Hildebrand et al., 2005). Later approaches are mostly based on a data set made available in the 2007 StatMT workshop (Koehn and Schroeder, 2007), and have attempted to use monolingual (Civera and Juan, 2007; Bertoldi and Federico, 2009) or comparable (Snover et al., 2008) corpus resources. These papers all show small, but significant, gains in performance when moving from Parliament domain to News domain.

3 Data

Our source domain is European Parliament proceedings (`http://www.statmt.org/europarl/`). We use three target domains: the News Commentary corpus (News) used in the MT Shared task at ACL 2007, European Medicines Agency text (Emea), the Open Subtitles data (Subs) and the PHP technical document data, provided as part of the OPUS corpus `http://urd.let.rug.nl/tiedeman/OPUS/`). We extracted development and test sets from each of these corpora, except for *news* (and the source domain) where we preserved the published dev and test data. The "source" domain of Europarl has 996k sentences and 2130k words.) We count the number of words and sentences in the English side of the parallel data, which is the same for both language pairs (i.e. both French-English and German-English have the same English). The statistics are:

	Comparable		Tune	Test
	sents	words	sents	sents
News	35k	753k	1057	2007
Emea	307k	4220k	1388	4145
Subs	30k	237k	1545	2493
PHP	6k	81k	1007	2000

Dom	Most frequent OOV Words
News (17%)	behavior, favor, neighbors, fueled, neighboring, abe, wwii, favored, nicolas, favorable, zhao, ahmedinejad, bernanke, favorite, phelps, ccp, skeptical, neighbor, skeptics, skepticism
Emea (49%)	renal, hepatic, subcutaneous, irbesartan, ribavirin, olanzapine, serum, patienten, dl, eine, sie, pharmacokinetics, ritonavir, hydrochlorothiazide, erythropoietin, efavirenz, hypoglycaemia, epoetin, blister, pharmacokinetic
Subs (68%)	gonna, yeah, f...ing, s..., f..., gotta, uh, wanna, mom, lf, ls, em, b....h, daddy, sia, goddamn, sammy, tyler, bye, bigweld
PHP (44%)	php, apache, sql, integer, socket, html, filename, postgresql, unix, mysql, color, constants, syntax, sesam, cookie, cgi, numeric, pdf, ldap, byte

Table 1: For each domain, the percentage of target domain word tokens that are unseen in the source domain, together with the most frequent English words in the target domains that do not appear in the source domain. (In the actual data the subtitles words do not appear censored.)

All of these data sets actually come with *parallel* target domain data. To obtain comparable data, we applied to standard trick of taking the first 50% of the English text as English and the last 50% of the German text as German. While such data is more parallel than, say, Wikipedia, it is far from parallel.

To get a better sense of the differences between these domains, we give some simple statistics about out of vocabulary words and examples in Table 1. Here, for each domain, we show the percentage of words (types) in the target domain that are unseen in the Parliament data. As we can see, it is markedly higher in Emea, Subs and PHP than in News.

4 Dictionary Mining

Our dictionary mining approach is based on Canonical Correlation Analysis, as used previously by (Haghighi et al., 2008). Briefly, given a multi-view data set, Canonical Correlation Analysis is a technique to find the projection directions in each view so that the objects when projected along these di-

rections are maximally aligned (Hotelling, 1936). Given any new pair of points, the similarity between the them can be computed by first projecting onto the lower dimensions space and computing the cosine similarity between their projections. In general, using all the eigenvectors is sub optimal and thus retaining top eigenvectors leads to an improved generalizability.

Here we describe the use of CCA to find the translations for the OOV German words (Haghighi et al., 2008). From the target domain corpus we extract the most frequent words (approximately 5000) for both the languages. Of these, words that have translation in the bilingual dictionary (learnt from Europarl) are used as training data. We use these words to learn the CCA projections and then mine the translations for the remaining frequent words. The dictionary mining involves multiple stages. In the first stage, we extract feature vectors for all the words. We use context and orthographic features. In the second stage, using the dictionary probabilities of seen words, we identify pairs of words whose feature vectors are used to learn the CCA projection directions. In the final stage, we project all the words into the sub-space identified by CCA and mine translations for the OOV words. We will describe each of these steps in detail in this section.

For each of the frequent words we extract the context vectors using a window of length five. To overcome data sparsity issue, we truncate each context word to its first seven characters. We discard all the context features which co-occur with less than five words. Among the remaining features, we consider only the most frequent 2000 features in each language. We convert the frequency vectors into TFIDF vectors, center the data and then binarize the vectors depending on if the feature value is positive of not. We convert this data into word similarities using linear dot product kernel. We also represent each word using the orthographic features, with n-grams of length 1-3 and convert them into TFIDF form and subsequently turn them into word similarities (again using the linear kernel). Since we convert the data into word similarities, the orthographic features are relevant even though the script of source and target languages differ. Where as using the features directly rending them useless for languages whose script is completely different like Arabic and English. For each language we linearly combine the kernel matrices obtained using the context vectors and the orthographic features. We use incomlete cholesky decomposition to reduce the dimensionality of the kernel matrices. We do the same preprocessng for all words, the training words and the OOV words. And the resulting feature vectors for each word are used for learning the CCA projections

waste	blutdruckabfall	0.274233
bleeding	blutdruckabfall	0.206440
stroke	blutdruckabfall	0.190345
dysphagia	dysphagie	0.233743
encephalopathy	dysphagie	0.215684
lethargy	dysphagie	0.203176
ribavirin	ribavirin	0.314273
viraferonpeg	ribavirin	0.206194
bioavailability	verfgbarkeit	0.409260
xeristar	xeristar	0.325458
cymbalta	xeristar	0.284616

Table 2: Random unseen Emea words in German and their mined translations.

Since a word can have multiple translations, and that CCA uses only one translation, we form a bipartite graph with the training words in each language as nodes and the edge weight being the translation probability of the word pair. We then run Hungarian algorithm to extract maximum weighted bipartite matching (Jonker and Volgenant, 1987). We then run CCA on the resulting pairs of the bipartite matching to get the projection directions in each language. We retain only the top 35% of the eigenvectors. In other relevant experiments, we have found that this setting of CCA outperforms the baseline approach.

We project all the frequent words, including the training words, in both the languages into the lower dimensional spaces and for each of the OOV word return the closest five points from the other language as potential new translations. The dictionary mining, viewed subjectively and intrinsically, performs quite well. In Table 2, we show four randomly selected unseen German words from Emea (that do not occur in the Parliament data), together with the top three translations and associated scores (which are *not* normalized). Based on a cursory evaluation of 5 randomly selected words in French and German

by native speakers (not the authors!), we found that 8/10 had correct mined translations.

5 Integration into MT System

The output of the dicionary mining approach is a list of pairs (f, e) of foreign words and predicted English translations. Each of these comes with an associated score. There are two obvious ways to integrate such a dictionary into a phrase-based translation system: (1) Provide the dictionary entries as (weighted) "sentence" pairs in the parallel corpus. These "sentences" would each contain exactly one word. The weighting can be derived from the translation probability from the dictionary mining. (2) Append the phrase table of a baseline phrase-based translation model trained only on source domain data with the word pairs. Use the mining probability as the phrase translation probabilities.

It turned out in preliminary experiments (on German/Emea) that neither of these approaches worked particularly well. The first approach did not work at all, even with fairly extensive hand-tuning of the sentence weights. It often hurt translation performance. The second approach did not hurt translation performance, but did not help much either. It led to an average improvement of only about 0.5 Bleu points, on development data. This is likely because weight tuning tuned a single weight to account for the import of the phrase probabilities across both "true" phrases as well as these "mined" phrases.

We therefore came up with a slightly more complex, but still simple, method for adding the dictionary entries to the phrase table. We add *four* new features to the model, and set the plain phrase-translation probabilities for the dictionary entries to zero. These new features are:

1. The dictionary mining translation probability. (Zero for original phrase pairs.)
2. An indicator feature that says whether *all* German words in this phrase pair were seen in the source data. (This will always be true for source phrases and always be false for dictionary entries.)
3. An indicator that says whether *all* German words in this phrase pair were seen in target data. (This is *not* the negation of the previous feature, because there are plenty of words in the target data that had also been seen. This feature might mean something like "trust this phrase pair a lot.")
4. The conjunction of the previous two features.

Interestingly, only adding the first feature was not helpful (performance remained about 0.5 Bleu points above baseline). Adding only the last three features (the indicator features) alone did not help at all (performance was roughly on par with baseline). Only when all four features were included did performance improve significantly. In the results discussed in Section 6.2, we report results on test data using the combination of these four features.

6 Experiments

In all of our experiments, we use two trigram language models. The first is trained on the Gigaword corpus. The second is trained on the English side of the target domain corpus. The two language models are traded-off against each other during weight tuning. In all cases we perform parameter tuning with MERT (Och, 2003), and results are averaged over three runs with different random initializations.

6.1 Baselines and Oracles

Our first set of experiments is designed to establish baseline performance for the domains. In these experiments, we built a translation model based *only* on the Parliament proceedings. We then tune it using the small amount of target-domain tuning data and test on the corresponding test data. This is row BASELINE in Table 3. Next, we build an oracle, based on using the *parallel* target domain data. This system, OR in Table 3 is constructed by training a system on a mix of Parliament data and target-domain data. The last line in this table shows the percent improvement when moving to this oracle system. As we can see, the gains range from tiny (4% relative Bleu points, or 1.2 absolute Bleu points for news, which may just be because we have more data) to quite significant (73% for medical texts).

Finally, we consider how much of this gain we could possible hope to realize by our dictionary mining technique. In order to estimate this, we take the OR system, and remove any phrases that contain source-language words that appear in *neither*

		BLEU				Meteor			
		News	Emea	Subs	PHP	News	Emea	Subs	PHP
German	BASELINE	23.00	26.62	10.26	38.67	34.58	27.69	15.96	24.66
	ORACLE-OOV	23.77	33.37	11.20	39.77	34.83	30.99	17.03	25.82
	ORACLE	24.62	42.77	11.45	41.01	35.46	36.40	17.80	25.85
French	BASELINE	27.30	40.46	16.91	28.12	37.31	35.62	20.61	20.47
	ORACLE-OOV	27.92	50.03	19.17	29.48	37.57	39.55	21.79	20.91
	ORACLE	28.55	59.49	19.81	30.15	38.12	45.55	23.52	21.77
ORACLE-OOV CHANGE		+2%	+24%	+11%	+5%	+0%	+12%	+6%	+7%
ORACLE CHANGE		+4%	+73%	+15%	+2%	+2%	+29%	+13%	+6%

Table 3: Baseline and oracle scores. The last two rows are the change between the baseline and the two types of oracles, averaged over the two languages.

	German		French	
	BLEU	**Meteor**	**BLEU**	**Meteor**
News	23.80	35.53	27.66	37.41
	+0.80	*+0.95*	*+0.36*	*+0.10*
Emea	28.06	29.18	46.17	37.38
	+1.44	*+1.49*	*+1.51*	*+1.76*
Subs	10.39	16.27	17.52	21.11
	+0.13	*+0.31*	*+0.61*	*+0.50*
PHP	38.95	25.53	28.80	20.82
	+0.28	*+0.88*	*+0.68*	*+0.35*

Table 4: Dictionary-mining system results. The italicized number beneath each score is the improvement over the BASELINE approach from Table 3.

the Parliament proceedings *nor* our list of high frequency OOV terms. In other words, if our dictionary mining system found as-good translations for the words in its list as the (cheating) oracle system, this is how well it would do. This is referred to as OR-OOV in Table 3. As we can see, the upper bound on performance based only on mining unseen words is about halfway (absolute) between the baseline and the full Oracle. Except in news, when it is essentially useless (because the vocabulary differences between news and Parliament proceedings are negligible). (Results using Meteor are analogous, but omitted for space.)

6.2 Mining Results

The results of the dictionary mining experiment, in terms of its effect on translation performance, are shown in Table 4. As we can see, there is a modest improvement in Subtitles and PHP, a markedly large improvement in Emea, and a modest improvement in News. Given how tight the ORACLE results were to the BASELINE results in Subs and PHP, it is quite impressive that we were able to improve performance as much as we did. In general, across all the data sets and both languages, we roughly split the difference (in absolute terms) between the BASELINE and ORACLE-OOV systems.

7 Discussion

In this paper we have shown that dictionary mining techniques can be applied to mine unseen words in a domain adaptation task. We have seen positive, consistent results across two languages and four domains. The proposed approach is generic enough to be integrated into a wide variety of translation systems other than simple phrase-based translation.

Of course, unseen words are not the only cause of translation divergence between two domains. We have not addressed other issues, such as better estimation of translation probabilities or words that change word sense across domains. The former is precisely the area to which one might apply domain adaptation techniques from the machine learning community. The latter requires significant additional work, since it is quite a bit more difficult to spot foreign language words that are used in new senses, rather that just never seen before. An alternative area of work is to extend these results beyond simply the top-most-frequent words in the target domain.

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for MT evaluation with improved correlation with human judgments. In *In Proceedings of Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at ACL*.

Nicola Bertoldi and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In *StatMT '09: Proceedings of the Fourth Workshop on Statistical Machine Translation*.

John Blitzer and Hal Daumé III. 2010. Domain adaptation. Tutorial at the International Conference on Machine Learning, `http://adaptationtutorial.blitzer.com/`.

Jorge Civera and Alfons Juan. 2007. Domain adaptation in statistical machine translation with mixture modelling. In *StatMT '07: Proceedings of the Second Workshop on Statistical Machine Translation*.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*.

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 *European Association for Machine Translation*.

H. Hotelling. 1936. Relation between two sets of variables. *Biometrica*, 28:322–377.

J. Jiang. 2008. A literature survey on domain adaptation of statistical classifiers. Available at `http://sifaka.cs.uiuc.edu/jiang4/domain_adaptation/survey`.

R. Jonker and A. Volgenant. 1987. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *StatMT '07: Proceedings of the Second Workshop on Statistical Machine Translation*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*.

Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, July.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 311–318.

Matthew Snover, Bonnie Dorr, and Richard Schwartz. 2008. Language and translation model adaptation using comparable corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.