Domain Adaptation in Machine Translation: Final Report

Marine Carpuat[1], Hal Daumé III[2], Alexander Fraser[3], Chris Quirk[4]
Fabienne Braune[3], Ann Clifton[5], Ann Irvine[6], Jagadeesh Jagarlamudi[2]
John Morgan[7], Majid Razmara[5], Aleš Tamchyna[8]
Katharine Henry[9], Rachel Rudinger[10]

marine.carpuat@cnrc-nrc.gc.ca, me@hal3.name, fraser@ims.uni-stuttgart.de, chrisq@microsoft.com
fabienne.braune@ims.uni-stuttgart.de, aca69@sfu.ca, anni@jhu.edu, jags@umiacs.umd.edu
john.j.morgan50.civ@mail.mil, razmara@sfu.ca, a.tamchyna@gmail.com
katiebethhenry@gmail.com, rachel.rudinger@yale.edu

[1]National Research Council Canada [2]University of Maryland, College Park [3]University of Stuttgart
[4]Microsoft Research [5]Simon Fraser University [6]Johns Hopkins University [7]Army Research Lab
[8]Charles University [9]University of Chicago [10]Yale University

December 24th, 2012

Contents

	Old Domain	New Domain (Medical)
Original German text	wenn das geschieht, würden die serben aus dem nordkosovo wahrscheinlich ihre eigene unabhängigkeit erklären.	darreichungsform : weißes pulver und klares , farbloses lösungsmittel zur herstellung einer injektionslösung
Human translation	if that happens, the serbs from north kosovo would probably have their own independence.	pharmaceutical form : white powder and clear , colourless solvent for solution for injection
SMT output	if that happens, it is likely that the serbs of north kosovo would declare their own independence.	darreichungsform : white powder and clear , pale solvents to establish a injektionslösung

Figure 1: Figure: Output of a SMT system. The left example is from the system's old training domain, the right is from an unseen new domain. Incorrect translations are highlighted in red, the two German words are unknown to the system, while the two English words are incorrect word sense problems.

1 Introduction

Statistical machine translation (SMT) systems perform poorly when applied on new domains. This degradation in quality can be as much as one third of the original systems performance; the figure 1 provides a small qualitative example, and illustrates that unknown words (copied verbatim) and incorrect translations are major sources of errors. When parallel data is plentiful in a new domain, the primary challenge becomes that of scoring good translations higher than bad translations. This is often accomplished using either mixture models that downweight the contribution of old domain corpora, or by subsampling techniques that attempt to force the translation model to pay more attention to new domain-like sentences. A more sophisticated approach recently demonstrated that phrase-level adaptation can perform better (Foster et al., 2010). However, these approaches are still less sophisticated than state-of-the-art domain adaptation (DA) techniques from the machine learning community (Blitzer & Daumé III, 2010). Such techniques have not been applied to SMT, likely due to the mismatch between SMT models and the classification setting that dominates the DA literature. The Phrase Sense Disambiguation (PSD) approach to translation (Carpuat & Wu, 2007), which treats SMT lexical choice as a classification task, allows us to bridge this gap. In particular, classification-based DA techniques can be applied to PSD to improve translation scoring. Unfortunately, this is not enough when only comparable data exists in the new domain. Here, we face the additional challenge of identifying unseen words and also unknown word senses of seen words and attempting to figure out potential translations for these lexical entries. Once we have identified potential translations, we still need to score them, and the techniques we developed for addressing the case of parallel data directly apply.

1.1 Goals

1. Understand domain divergence in parallel data and how it affects SMT models, through analysis of carefully defined test beds that will be released to the community.

2. Design new SMT lexical choice models to improve translation quality across domains in two settings:

   (a) When new domain parallel data is available, we leverage existing machine learning algorithms to adapt PSD models, and explore a rich space of context features.

   (b) When we only have comparable data in the new domain, we will learn training examples for PSD by identifying new translations for new senses.

1.2 Approach

While BLEU scores suggest that SMT lexical choice is often incorrect outside of the training domain, previous work does not yet fully identify the sources of translation error for different domains, languages and data conditions. In a preliminary analysis in a DA setting without new parallel data, we have identified unseen words and senses as the main sources of error in many new domains, by analyzing impacts on BLEU. We conduct similar analyses for the setting

with new parallel data. We also consider sources of error like word alignment or decoding. We exploit parallel text to better understand differences between general and domain-specific phrase usage (Foster et al., 2010), and their impact on SMT.

We can learn differences between general language terms, domain-specific terms, and domain-specific usages of general terms, by using their translations as a sense annotation. This is a complex task, since domain shifts are not the only cause of translation ambiguity. For instance, in English to French translation, run is usually translated in the computer domain as xcuter, and in the sports domain as courir; but other senses (such as diriger, to manage) can appear in many domains. Sense distinctions also depend on language pairs, which suggests that comparable data in the input language truly is necessary. For example, consider the English words virus and window. When translating into French, regardless of whether one is in a general domain or a computer domain, they are translated the same way: as virus and fenłtre, respectively. However, when translating into Japanese, the domain matters. In a general domain, they are respectively translated as and ł; but in a computer domain they are transliterated.

To build SMT systems that are adapted to a new domain, we first consider the setting with parallel data from the new domain. We build on a translation approach that explicitly models the domain-specificity of phrase pair types to re-estimate translation probabilities (Foster et al., 2010). Rather than using static mixtures of old and new translation probabilities, this approach learns phrase-pair specific mixture weights based on a combination of features reflecting the degree to which each old-domain phrase pair belongs to general language (e.g., frequencies, centrality of old model scores), and its similarity to the new domain (e.g., new model scores, OOV counts). By moving to a PSD translation model, we can attempt much more sophisticated adaptation, and better model the entire spectrum between general and domain specific senses. In PSD, based on training data extracted from word-aligned parallel data, a classifier scores each phrase-pair in the lexicon, using evidence from the input-language context. Although there are certainly non-lexical affects of domain shift, we focus on the lexicon, which is the most fruitful target given our past experience.

With parallel data, our work focuses on adapting PSD to new domains in order to learn better scores for lexical selection. We design adaptation algorithms for PSD, by applying existing learning techniques for DA (Blitzer & Daumé III, 2010). Such approaches typically have two goals: (1) to reduce the reliance of the learned model on aspects that are specific to the old domain (and hence are unavailable at test time), and (2) to use correlations between related old-domain examples and new-domain examples to port parameters learned on the old to the new domain. Such techniques can be directly applied to the PSD translation model, using large context as features. We consider local contexts features like in past work (Carpuat & Wu, 2007), but our approach can leverage much larger contexts (the paragraph, or perhaps the entire document (Carpuat, 2009)) to build better models, as well as morphological features (Fraser et al., 2012) to tackle the data sparsity issues that arise when dealing with small amounts of new domain data.

With only comparable text, we must spot phrases with new senses, identify their translations, and learn to score them. We attack the identification challenge using context-based language models (n-gram or topic models) to identify new usages. For example, in the computer domain, one can observe that window still appears on the English side, but ł (the general domain word for window) has disappeared in Japanese, indicating a potential new sense. For identifying translations we study dictionary mining (Daumé III & Jagarlamudi, 2011) or active learning (Bloodgood & Callison-Burch, 2010). The scoring problem can be addressed exactly as before. While finding new senses and translations is a challenging problem even in a single domain, we believe that differences that might get lost in a single domain with plentiful data are more apparent in an adaptation setting.

1.3 Evaluation

We create standard experimental conditions for domain adaptation in SMT and make all resources available to the community. We consider three very different domains with which we have past experience: medical texts, movie subtitles (Daumé III & Jagarlamudi, 2011) and scientific texts. We focus on French-English data, since our team includes native speakers of these two languages. We evaluate the performance of all adapted and non-adapted translation systems using standard automatic metrics of translation quality such as BLEU and Meteor. However, we show that these generic metrics do not adequately capture the impact of adaptation on domain-specific vocabulary, and we investigate how to evaluate domain-specific translation quality in a more directly interpretable way. We study lexical choice accuracy (automatically checking whether a translation predicted by PSD using source context is correct) using gold standard annotations. We evaluate extracting this knowledge by manually correcting automatic word-alignments and also by using terminology extraction techniques (e.g., finding translations of the keywords in scientific texts, etc).

1.4 Summary

Domain mismatch is a significant challenge for statistical machine translation. Our work contributes to elucidating this problem through careful data analysis, provides test beds for future research, explores the gap between statistical domain adaptation and statistical machine translation, and improves translation quality through novel methods for identifying new senses from comparable corpora.

2 Data Analysis

We chose French-English as a test-bed language pair mostly because of the availability of data in a number of domains, and the relative efficacy of standard translation methods. That is, we believe that SMT systems work pretty well in this domain, so translation failures during domain shift should be attributed more to domain issues than problems with the SMT system. There are downsides to this efficacy, however: a system that learns efficiently can also adapt more quickly, making adaptation more challenging.

Four major domains are at play:

- **Hansard**: Canadian parliamentary proceedings. This large corpus consists of manual transcriptions and translations of meetings of Canada's House of Commons and its committees[1] from 2001 to 2009. Discussions cover a wide variety of topics, and speaking styles range from prepared speeches by a single speaker to more interactive discussions.
- **EMEA**: Documents from the European Medicines Agency, made available with the OPUS corpora collection (Tiedemann, 2009). This corpus primarily consists of drug usage guidelines.
- **News**: News commentary corpus made available for the WMT 2009 evaluation[2]. It has been commonly used in the domain adaptation literature (Koehn & Schroeder, 2007; Foster & Kuhn, 2007; Haddow & Koehn, 2012, for instance).
- **Science:**: Parallel abstracts from scientific publications in many disciplines including physics, biology, and computer science. We collected data from two distinct sources: (1) Canadian Science Publishing[3] made available translated abstracts from their journals which span many research disciplines; (2) parallel abstracts from PhD theses in Physics and Computer Science collected from the HAL public repository (Lambert et al., 2012).
- **Subs:** Translated movie subtitles, available through the OPUS corpora collection (Tiedemann, 2009). In contrast with the other domains considered, subtitles consists of informal noisy text.

	Hansard		EMEA		Science		Subs	
	French	English	French	English	French	English	French	English
Sentences	8,107,356		472,231		139,215		19,239,980	
Tokens	161,695,309	144,490,268	6,544,093	5,904,296	4,292,620	3,602,799	154,952,432	174,430,406
Types	191,501	186,827	34,624	29,663	117,669	114,217	361,584	293,249

Table 1: Basic characteristics of the training data in each domain.

	French types	English types	Pair types	French tokens	English tokens	Pair tokens
Hansard∩EMEA	17,845	13,743	63,087	6,124,518	5,522,972	6,290,162
EMEA−Hansard	16,779	15,920	431,877	419,575	381,324	2,002,943
Hansard∩Science	40,016	32,947	135,247	4,057,191	3,358,471	3,995,699
Science−Hansard	77,653	81,270	879,423	235,429	244,328	1,179,428
Hansard∩Subs	98,048	68,274	694,212	152,519,138	171,806,360	199,375,051
Subs−Hansard	263,536	224,975	6,471,868	2,433,294	2,624,046	18,649,558

Table 2: Differences between domains.

[1] http://www.parl.gc.ca
[2] http://www.statmt.org/wmt09/translation-task.html
[3] http://www.nrcresearchpress.com

2.1 SMT quality across domains

In order to get a better understanding of differences between domain, we compare the translation quality of SMT systems when translating in domain, out of domain, and using simple adaptation techniques that combined data from both domains.

First, we compare BLEU scores obtained on test sets from each of the NEW domain for phrase-based SMT systems trained under 3 distinct data conditions: (1) on OLD domain data only (Canadian Hansard), (2) on NEW domain data only (News, Medical, Science and Subtitles), and (3) on the concatenation of OLD + NEW data. Table 3 shows that BLEU score drops significantly when testing on out-of-domain data for three of the four domains considered: the Hansard trained system yields scores that are 7 to 12 points lower than the in-domain systems. The results are different for the News domain: the Hansard system actually translates News data with a better BLEU score than the system trained on News. This can be explained by the small amount of parallel data available to train the News only system, and the fact that the News corpus is much closer to the Hansard than any of the other domains considered.

Training Domain	News	EMEA	Science	Subtitles
OLD	22.61	22.72	21.22	13.64
NEW	20.33	34.83	32.49	20.57
OLD+NEW	23.82	34.76	30.17	20.41

Table 3: BLEU scores for phrase-based Moses evaluated in each NEW domain: translation quality almost always degrades significantly when translating out of domain, and simply concatenating data from different domains does not help.

3 Analysis of Problems Caused by Shifting Domain

3.1 Baselines

The moses decoder and its experiment management system were used to train, tune, and test baseline systems. The following baselines are meant to reflect the best possible performance of a system without adaptation. The OLD domain is always text from the Canadian Hansard. Tuning is always performed on a held out set extracted from the NEW domain. Tuning was always performed with batch MIRA. Training of the language model was performed on either the target side of the entire parallel training data or only on the text from the NEW domain in the parallel training data. All language models contain 5 grams and use knesser-ney smoothing.

3.1.1 OLD Domain System on NEW Domains with No Tuning

The following table shows the BLEU scores obtained by decoding with a system trained exclusively on data from the OLD domain and tested on data from each of the NEW domains. These scores are intended to show the performance of a system that has not been exposed to in-domain data.

Domain	BLEU Score
Hansard	40.69
News	22.61
Medical	20.90
Science	19.38
Subtitles	12.48

Table 4: BLEU scores of the baseline system without tuning. Language models were trained on the English side of the Hansard corpus. During training the system was not exposed to data from the NEW domains.

The above table clearly indicates that moving to a new domain can affect the performance of a statistical machine translation system. What is the source of the change in performance? Later we will attempt to answer this question by analyzing different kinds of errors that occur in smt.

3.1.2 Tune on In-domain Data

The following table shows the BLEU scores that result from training on OLD domain data and tuning on data from the NEW domain. Modified tuning and test sets were generated that were restricted to segments that were not "seen" in the training data. Subsequent domain adaptation work will assume a small corpus of NEW domain data exists for tuning, thus scores from those systems should achieve at least the scores given in table 5.

Domain	BLEU Score
Hansard	41.54
News	23.82
Medical	28.69
Science	26.13
Subtitles	15.10

Table 5: BLEU scores from Old domain trained and NEW domain tuned systems.

3.2 Error Analysis

Next we investigate the source of decreased BLEU scores when moving to a new domain. In the following investigation we make two key assumptions:

1. Enough parallel data is available in the new domain for tuning and testing.
2. Enough monolingual data is available in the target language of the new domain for training a language model.

Other sections of this report will consider the case where comparable data is available in the NEW domain. We consider four kinds of errors:

SEEN: new words in the new domain,

SENSE: new, new-domain specific, translations for known words,

SCORE: wrong preference for non-new-domain translations, and,

SEARCH: search algorithm chooses incorrect word.

3.2.1 SEEN Errors

For errors of type SEEN we conduct the following experiment. We build "augmented" phrase and reordering tables by adding the unseen words and phrases to the tables trained on only the OLD domain data. The resulting tables are tuned and tested on the same data from the NEW domain that is used to test and tune the OLD system. The gap between the BLEU scores for the OLD and augmented systems indicates the improvement that can be gained by methods for automatically discovering corrections for unseen errors. Compare table 6 to table 5 to find the gap.

domain	augmented
News	23.87
Medical	31.02
Science	27.72
Subtitles	15.91

Table 6: Analysis of seen errors. Unseen words and phrases were added to the OLD system's phrase table and reordering table.

3.2.2 SENSE Errors

For SENSE errors we perform the same kind of experiments as we did for SEEN errors except that we augment the translation tables with translation pairs containing new senses. By definition a phrase has a new sense if it appears in both the OLD and NEW domains on the source side language but its translations in the target language are different in the OLD and NEW domains.

Again, compare these scores with those given in table 5 to find the gap.

domain	augmented
News	23.95
Medical	30.59
Science	27.29
Subtitles	16.41

Table 7: Analysis of sense errors. Words and phrases with new senses were added to the OLD system's phrase table and reordering table.

3.2.3 SCORE Errors

To access score errors we run different kinds of experiments than the ones we ran for seen and sense errors. Instead of augmenting tables, we considered the phrase pairs that were in both the OLD and NEW domain tables. The feature scores came from either the OLD table or the NEW table. One system that we called "score old" was built with the scores from the OLD system. The other system we called "score new" and was built with scores from the NEW table. These experiments involved the following steps:

1. Train a system on data from the OLD domain

2. Train a system on data from the NEW domain
3. Intersect the phrase pairs from the phrase tables from the systems built in steps 1 and 2 above
4. Build a "score old " system by inserting scores from the phrase-pairs given in the system built in step 1
5. Build a "score new " system by inserting scores from the phrase pairs given in the system built in step 2
6. Tune and test the systems built in the previous 2 steps on data from the NEW domain

The results in table 8 were obtained from systems with tables containing phrases in both the OLD and NEW domain system tables and feature scores from the OLD domain table.

domain	SCORE OLD	SCORE NEW
News	22.80	22.22
Medical	29.23	30.23
Science	26.21	28.98
Subtitles	14.99	16.25

Table 8: Analysis of score errors. Tables trained on the OLD and NEW domains were intersected. The numbers in the SCORE OLD column are scores that were obtained by the system trained on the OLD data. The numbers in the SCORE NEW column are scores that were obtained by the system trained on the NEW data.

Domain	BASE	SEEN	SENSE
News	23.82	23.87	23.95
Medical	28.69	31.02	30.59
Science	26.13	27.72	27.29
Subtitles	15.10	15.91	16.41

Table 9: BLEU scores summarizing the results of adding SEEN and SENSE errors to the OLD system.

Error Analysis Conclusions The results shown in this section are summarized in tables 9 and 8. The gap between the scores in columns 2 and 3 of table 8 shows the impact of errors attributed to incorrect feature scores when moving to a new domain. The scores in columns 3 and 4 of table 9 shows the impact of errors of type SEEN and SENSE when moving to a new domain. All these results demonstrate that moving to a new domain has a large impact on the performance of an SMT system and that errors of type SEEN, SENSE, and SCORE occur in the four domains we considered. We hoped to show that errors of one type stood out as more severe than the others, but at least for the phrase-based SMT systems studied in this work, this was not the case. Errors of type SCORE actually decreased when moving to the News domain. Even if we exclude the News domain, errors of type SEEN and SENSE have different impacts in the other domains. Errors of type SENSE are higher for the Subtitles domain while they are lower for the Medical and Science domains.

4 Micro-level Evaluation: WADE Analysis

Section 3 presented a macro-level study of how several error types affect translation performance (in our case, measured by BLEU). In this section, we present a *micro-level* evaluation tool for studying the same error types. We call this technique WADE, or Word Alignment Driven Evaluation. WADE identifies errors on the sentence level in real translation output, and we have developed a visualization tool for browsing error-tagged machine translation output. In addition to sentence-level visualizations, we present aggregate statistics over all sentences in a test set.

4.1 WADE Analysis

Our WADE technique analyzes MT system output at the word level, allowing us to (1) manually browse visualizations of data annotated with error types, and (2) aggregate counts of errors. WADE is based on the fact that we can automatically word-align a test set French sentence and its reference English translation, and we can use the MT decoder's word alignments between a test set French sentence and its machine translation. We can then check whether our translated sentence has the same set of English words aligned to each French word that we would hope for, given the English reference. WADE's unit of analysis is a word alignment between test set French words and their reference translations.

Based on the word-aligned machine translation, we automatically annotate each test-reference word alignment with one of the following categories:

- Correct
- OOV-Freebie
- Sense-Freebie
- Score/Search Error
- OOV-Wrong
- New Sense-Wrong

We determine whether French words are out-of-vocabulary (OOV) or not by looking at the French side of the parallel training data. We determine whether English translations of French words are new senses or not by looking at the word-aligned parallel training data (i.e. the lexical t-table). OOV-Freebies are situations in which the correct translation for an OOV French word is its identity (e.g. many person and place names are identical in French and English). Sense-freebies are situations in which the correct translation for a French word is its identity, but we had seen the French word translated *as something else* in the t-table[4]. When our MT system encounters a French word that it does not know how to translate, its default behavior is to copy the word in the output. In both 'freebie' cases, the copied word is correct. OOV-Wrong annotations occur when a French word is OOV but the identical translation is incorrect. New Sense-Wrong annotations occur when the reference English translation of a French word is new. When the MT system has access to the correct English translation of a French word but makes either a search or score error and does not produce the correct translation.

4.2 Results

Figures 2 and 3 show examples of the output from the WADE visualization tool that we have created.

WADE is fundamentally based upon word alignments, so alignment errors may affect its accuracy. Such errors are obvious in manually inspecting sentence triples using the visualizer. In developing this tool, we were particularly skeptical that alignment errors would make aggregate counts of the above annotations uninformative. In order to estimate how much alignment errors affect WADE, one the French speakers on our team manually inspected the word alignments for 1,088 French-English test set sentences in the EMEA domain. Our annotator marked 133 sentences (or 12% of the data she inspected) as bad translation pairs and manually corrected the automatic word alignments in the remaining 955 sentences.

[4]These cases are likely a result of a bad word alignment. Note also that, in these cases, the MT system does not translate the French word as its previously observed English sense because either the unigram lexical translation rule was not extracted by the grammar or it was pruned from the grammar that we used in decoding.

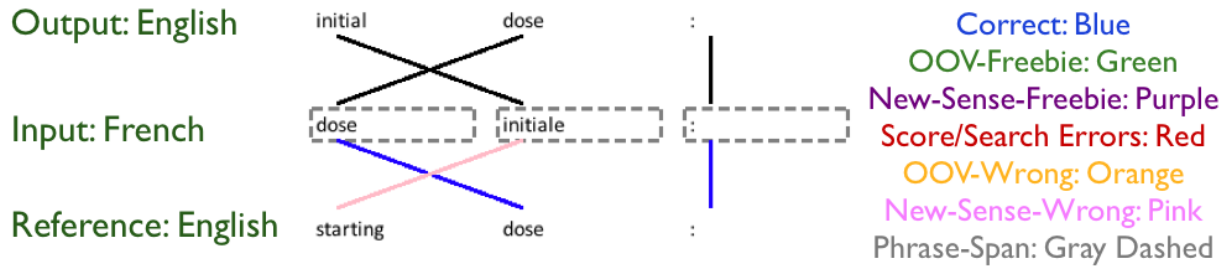


Figure 2: Example of WADE visualization

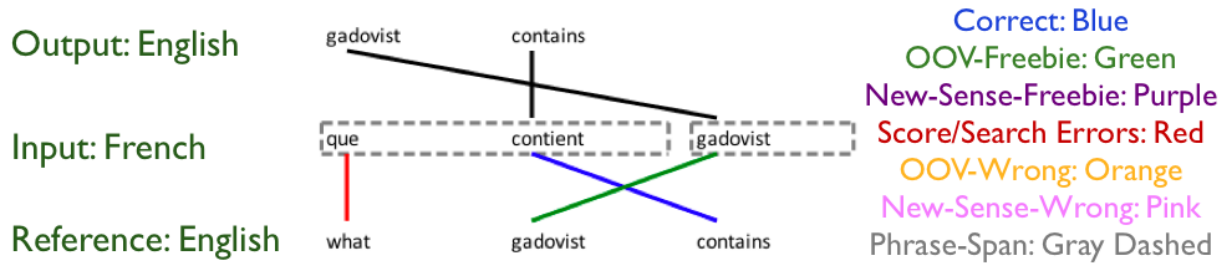


Figure 3: Example of WADE visualization

Figures 4 and 5 show WADE analyses for several experimental outputs in the EMEA domain. In each figure, pairs of bars correspond to analyses of MT output from systems trained on the following datasets: (1) Hansard domain data only, (2) EMEA domain data only, (3) concatenation of Hansard and EMEA data.

There is no clear trend in the comparison between the analyses based on automatic alignments and the analyses based on manual alignments. In the Hansard-only-train experimental condition, the analysis based on manual alignments reports fewer errors overall than the one based on automatic alignments. However, in the other two conditions, the analyses based on manual alignments report slightly more errors overall. Although it would be nice to see more consistency, the rank order between the experimental conditions is the same for both sets of alignments. That is, both report that the system trained on Hansard data alone is the worst performer and the system trained on the concatenation of the two datasets is the best performer.

In nearly all experimental conditions, score and search errors (labeled *incorrect*) make up the majority of errors, followed by new sense errors and then seen (OOV) errors. However, interestingly, the system trained on Hansard domain data only makes both more new sense errors and more seen (OOV) errors than the systems that also make use of in-domain training data. It makes only slightly more score and search errors. This means that the performance degradation that we observe when shifting domains can be attributed to words with new senses and unseen (OOV) words more so than score and search errors.

4.3 WADE Conclusions

Our aggregate WADE results support the conclusions made through the macro-level analysis presented in Section 3. That is, WADE shows that sense and seen errors account for more of the performance degradation that we observe in shifting domains than either score or search errors. Moreover, the WADE visualizer is an effective tool for browsing examples of all error types in real MT output.

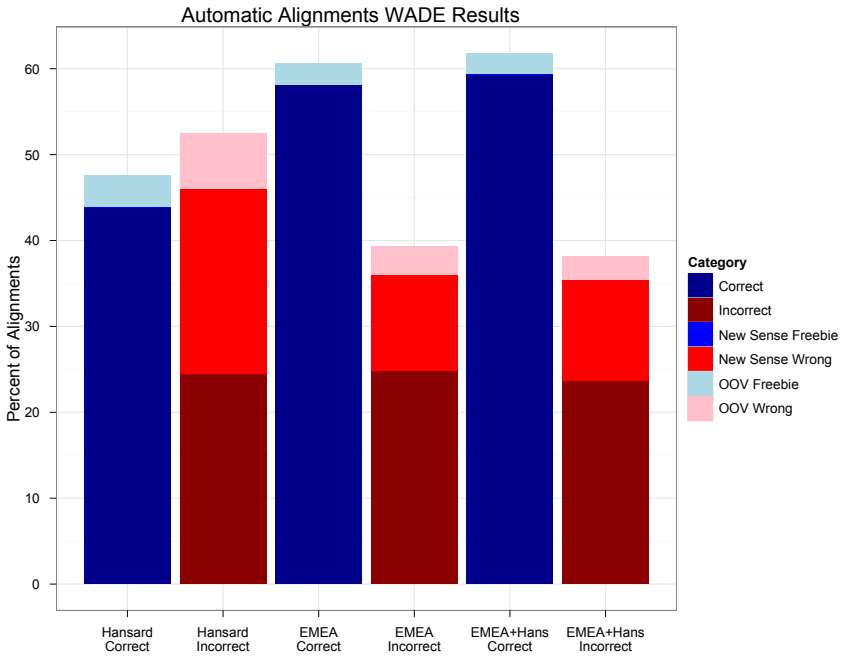


Figure 4: WADE results using automatic alignments. The three pairs of bars correspond to output from systems trained on the following datasets: (1) Hansard domain data only, (2) EMEA domain data only, (3) concatenation of Hansard and EMEA data.

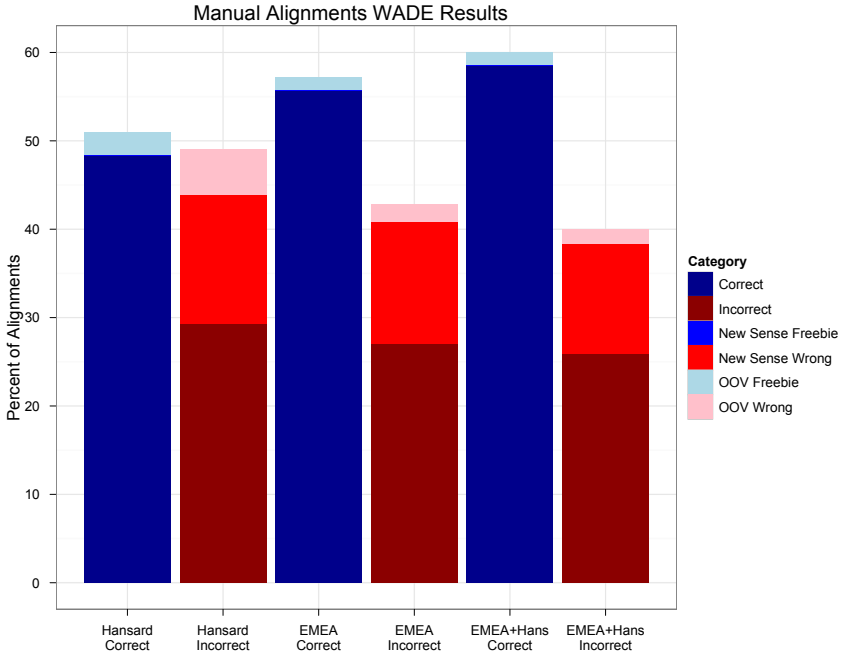


Figure 5: WADE results using manually corrected alignments. The three pairs of bars correspond to output from systems trained on the following datasets: (1) Hansard domain data only, (2) EMEA domain data only, (3) concatenation of Hansard and EMEA data.

5 Phrase Sense Disambiguation for Domain Adapted SMT

Our analysis of domain effects, which we covered in Section 3, shows that SMT performance degrades when translating out of domain because of different types of lexical choice errors: SEEN (out of vocabulary errors), SENSE (known words with unknown translation sense in the NEW domain) and SCORE (known words with known translations but different translation probability distribution in the OLD and NEW domains). Most approaches to domain adaptation in SMT rely on coarse uniform mixtures of OLD and NEW domain models. As a result, they do not directly target these finer-grained lexical phenomena, and yield small improvements in BLEU score 2.

We propose to tackle domain adaptation using **Phrase Sense Disambiguation (PSD)** modeling Carpuat & Wu (2007). PSD is a discriminative translation lexicon, which scores translation candidates for a source phrase using source context, unlike standard phrase-table translation probabilities which are independent of context.

5.1 Introduction to Phrase Sense Disambigation

PSD views phrase translation as a classification task. At test time, the PSD classifier uses source context to predict the correct translation of a souce phrase in the target language. At training time, PSD uses word alignment to extract training instances, exactly as in a standard phrase-based SMT system. However, the extracted training instances are not just phrase pairs, but occurrences of source phrases **in context** annotated with their English translations.

5.2 Vowpal Wabbit for PSD (and other NLP tasks)

We chose to use Vowpal Wabbit, implemented by John Langford, to implement PSD. Vowpal Wabbit (VW), has a fast implementation of stochastic gradient descent and L-BFGS for many different loss functions. VW was built into a library (for this workshop). It is very widely used for machine learning tasks.

It has built-in support for:

- Feature hashing (scaling to billions of features)
- Caching (no need to re-parse text)
- Different losses and regularizers
- Reductions framework to binary classification
- Multithreaded/multicore support

Our "weird" setting (for many machine learning researchers) is that we use label-dependent features. This is normal for NLP researchers.

Think of it like ranking. Here is a sample problem:
x = le croissant rouge
y1 = the red croissant
y2 = the croissant red
y3 = the croissant
y4 = the red
This could be another problem in the same data set:
x = mange
y1 = eat
y2 = eats
y3 = ate

Different inputs have different numbers and definitions of possible labels, each with it's own features. We define the feature space as the $X * Y$ cross-product and either:

1. Regress on loss (csoaa_ldf)
2. Use a classifier all-versus-all (wap_ldf)

For information on these two algorithms, see the VW documentation which is available from John Langford's VW web page at: http://hunch.net/~vw/

5.3 Integrating VW in Phrase-based Moses

When developing PSD, we extended Moses in a number of ways. Most importantly, Moses can now be linked with the VW library and classifier predictions can be directly incorporated as features in the log-linear model. The overall architecture was designed to be simple and extensible. PSD itself is again a library. This allowed us to use the same code for training and decoding, avoiding code duplication and assuring consistent definition and configuration of features. A diagram of the library is shown in Figure 6.

The logic of feature generation is clearly separated from any VW specifics. The code that extracts and generates features simply gets an implementation of the abstract class *FeatureConsumer* — we provide 3 implementations. One generates features in text format for VW and stores them in an output file. The other two use VW directly via its library interface. In order to add different classifiers, such as MegaM, one only needs to implement this abstract class.

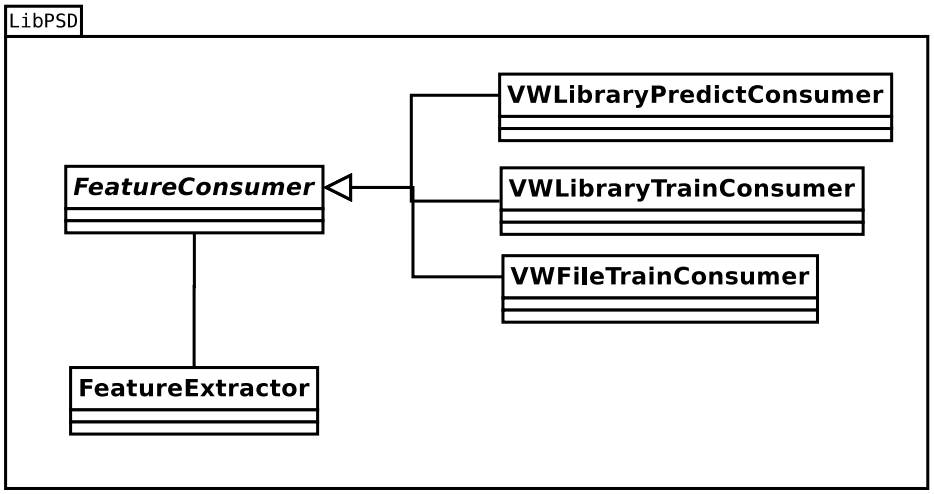


Figure 6: UML diagram of the PSD library.

The feature extractor contains implementations of various types of features for PSD. A configuration file (in .ini format) specifies which features should be enabled and sets their parameters (such as context length). Using the same configuration file in training and decoding guarantees that features will be consistent. Specifically, we implemented the following features:

- Source/target phrase indicator features.
- Source/target phrase-internal word features.
- Source context features. Values of defined factors in a limited context window.
- Source bag-of-words features.
- Score features. Cummulative quantized translation model log-scores.
- Indicator feature marking the most frequent translation.
- Paired features. Word pair indicator features based on word alignment.

During training, our modified version of the phrase extraction routine outputs information about each extracted phrase (sentence ID, position). This data is then used to construct training examples for VW (using the PSD library and the VW "file consumer") — along with the parallel corpus and (factored) annotation which includes lemmas and morphological tags. VW model is then trained. We parallelized each of these steps and achieved a considerable speed-up in training.

For decoding, we implemented a new feature function in Moses (*PSDScoreProducer*). This feature function evaluates all translation options of a given source span at the same time by querying VW for each of them, then inversely exponentiating the VW score (i.e. loss) and normalizing over all the options to get a probability distribution.

The feature score is stateless in the sense that it does not depend on the target side. On the other hand, it does require information about source context, and as such, it does not completely fit in the definition of stateless feature functions in Moses. Moreover, even stateless functions are evaluated during decoding in Moses (not ahead of time), which — aside from performance concerns — implies that the initial pruning of translation options is done without their scores.

We therefore integrated our feature in an ad-hoc manner. This allowed us to evaluate it before decoding of each sentence. Once translation options are collected from phrase tables, our function scores each of them. Then the initial pruning is done. During decoding (i.e. search for the best hypothesis), our feature function is not queried. Otherwise, PSD is a normal feature function. As such, it has a weight associated with it, which is optimized during tuning.

In terms of performance, Moses with PSD takes 80% relative longer than the Mose baseline without PSD, which is quite efficient. We made queries to VW thread-safe and tested all of our code in a massively parallel setting.

PSD is also fully integrated in Moses' Experiment Management System (EMS, experiment.perl) which allows potential new users to quickly create experiments with PSD.

All of our code is publicly available in the Moses repository in the branch *damt_phrase*.

We have also created another branch which integrates VW into the Moses implementation of hierarchical models, this branch is called *damt_hiero*. We have particularly focused on Hiero (Chiang, 2005). Please see section 8 for more details.

6 Intrinsic Lexical Choice

6.1 Task Overview

It has been observed that words acquire new senses and that the distribution of senses changes in different domains. For the purposes of this task two senses are distinct if they are translated differently. While BLEU allows to evaluate the overall quality of the translations, it does not directly examine how the system is doing at translating new and ambiguous senses. To address this we created an additional method of evaluation that consists of directly examining the accuracy of translation on phrases that are likely to change senses in our given domains. We call these phrases representative phrases. Such a metric helps us to identify how well the system is adapting to a new domain independent of its BLEU score, as well as to compare performance of a full-fledged MT system with output from a PSD classifier, which could help us to select productive features without running the full MT pipeline.

6.2 Selecting Representative Phrases

For the representative phrases we wanted to identify phrases that have multiple senses within either the new or the old domain as well as phrases that acquire new senses in the new domain.

We used a semi-automatic approach to identify representative phrases. We first used the phrase table from the Moses output to rank the phrases in each domain using TF-IDF scores with Okapi BM25 weighting in order to identify meaningful phrases in each of the domains. For each of the three new domains (EMEA, Science, and Subs), we found the interesect of phrases with the old and the new domain. We then looked at the different transaltions that each of these phrases had in the phrase table and a French speaker selected a subset of these phrases that have multiple senses.

In addition to the manually chosen phrases, we also identified words where the translation with the highest lexical weight varied in different domains, with the intuition being that these phrases were ones that were likely to have acquired a new sense. The top 600 phrases from this were added to the manually selected representative phrases to form a list of 812 representative phrases.

6.3 Creating the Gold Standard

Once the list of representative phrases was established, we created the gold standard for the intrinsic lexcial choice task as follows.

1. Extract phrase sense disambiguation files for all of the domains.
2. Filter the PSD files to only include representative phrases and their translations.
3. Create a list of distinct translations and the lexcial weight of that translation from the French to English lexical weight file from Moses.
4. For each representative phrase rank translations by decreasing lexical weight and filter the file to only include translations such that the lexical weight is greater than zero and the sum of lexical weights for that phrase is less that 0.8.
5. Filter the PSD files from step 2 to only include instances where the representative phrase was translated as one of the translations in step 4.

The resulting file is used as the gold standard for the intrinsic lexical choice task.

6.4 Effect of Multiple References

One of the disadvantages of our setup is that there is only one reference translation. As a result, there may be instances where multiple translations are possible, and the system output is correct but does not match the reference translation. One way that we can approximate having multiple reference sets is by calculating the Meteor score for the translations. Meteor aligns the system output to the reference translation using any combination of exact matches, stemming matches, synonym matches (using WordNet), and paraphrase matches (using a paraphrase table created from CCB word).

To examine the effect of having a single reference set when translating representative phrases, we looked at Meteor alignments between Moses output for the representative phrases and the reference translation on a system trained only on Hansard, only on EMEA, and on Hansard concatenated with EMEA using only exact matches, only stemming matches, only synonym matches, and only paraphrase matches. For all three systems, the majority of matches were made by exact alignments and for EMEA and Hansard + EMEA, a very small percent of matches were made through stemming, synonyms, or paraphrases. The only system where a significant portion of the output was aligned through either synonyms or paraphrases was on the Hansard trained system. In this case fewer matches were made through exact matches and instead the system relied more heavily on synonyms and paraphrases to align the data.

Table 10: Percent of Alignments Made for Representative Phrases

Trained On	Exact	Stemming	Synonym	Paraphrase
Hansard	78.02%	0.85%	1.86%	9.17%
EMEA	93.16%	0.68%	0.75%	0.86%
Hansard + EMEA	92.52%	0.45%	0.56%	1.92%

One concern was that distinct senses of a word may be counted as synonyms or paraphrases by Meteor, but may not actually be synonymous in context. For instance, *administration* can be translated as either 'administration' or 'directors' in some cases, but it would be incorrect to translate *voie d'administration* as 'route of directors'. We therefore had a French speaker annotated whether or not the alignment was correct in the context of the sentence. The precision of the alignments is reported in 10. Over all three systems there is high precision for synonym matches. In the EMEA trained system there is also high precision for paraphrase matches. However, only about half of the paraphrases made in the Hansard trained system were judged to be accurate paraphrases and the concatenated system falls in between the two.

Table 11: Precision of Meteor Representative Phrases Alignments

Trained On	Synonym	Paraphrase	Either
Hansard	0.98	0.47	0.50
EMEA	0.98	0.95	0.95
Hansard + EMEA	0.97	0.68	0.73

6.5 Summary

We developed this intrinisc lexical choice task as an alternative evaluation metric that measures how successfully a system translates new and ambiguous phrases in the target domain. This also allows us to compare performance between a word-sense disambiguation classifier and a full machine translation system. Ambiguous phrases were identified through a combination of TF-IDF weighting and identifying words whose lexical weights vary greatly between the source and target domain. Although these translations are based on a single reference translation, experiments with Meteor demonstrate that only a small percent of additional alignments are made through synonym or stemming matches, suggesting that the gains of having a second reference translation would be small. The intrinsic lexical choice task will allow us to evaluate how the system specifically performs on the domain adaptation task in a precise and informative way.

7 Phrase-based PSD

7.1 Baseline

Before experimenting with PSD, we carried out a brief evaluation of currently available tuning algorithms. We used one 16^{th} of Canadian Hansards data for training, the tuning and evaluation sets were also taken from Hansards. Each of the evaluated methods was run 5 times. Table 12 summarizes the achieved results. Batch MIRA outperformed MERT, PRO and their combination, while none of the other methods was significantly better than any of its competitors. We therefore used batch MIRA for tuning in all of our experiments.

Algorithm	BLEU	StDev
MERT	24.85	0.13
PRO+MERT	24.88	0.03
PRO	24.91	0.02
Batch MIRA	**25.04**	0.03

Table 12: Evaluation of tuning algorithms.

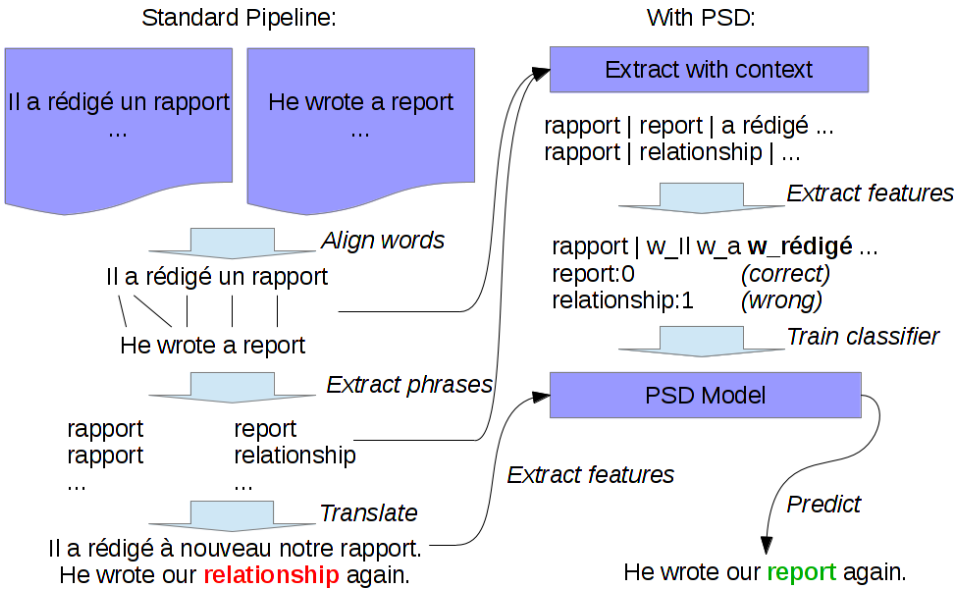


Figure 7: PSD pipeline in a phrase-based decoder.

7.2 Phrase-based PSD

In the phrase-based setting, phrase-sense disambiguation has potential to mitigate many of the inherent problems of this approach to MT. Specifically, by looking at wider context on the source side, we can make the task of lexical selection easier. Consider translation of "*rapport*" in the example in Figure 7. Even though the immediate context ("*notre*") would suggest that "*relationship*" is the correct translation, the word "*wrote*" makes the translation "*report*" much more likely. This word lies outside the scope of current state-of-the-art models, yet PSD can use it to infer the lexically correct translation.

Moreover, the generalization provided by this model could be beneficial when moving to new domains, even without applying any techniques for domain adaptation.

We ran experiments with PSD on various domains and data sizes. So far, we have not been able to improve the BLEU score. We are currently investigating the possible reasons for our results.

8 Soft Syntax and PSD for Hierarchical Phrase-Based SMT

Instead of using phrase sense disambiguation for domain adaptation (5) within a phrase-based SMT system (7), we propose to use word sense disambiguation as well as syntactic features within a hierarchical phrase-based SMT framework (Chiang, 2005). For this purpose, VW has been integrated in a hierarchical MT system. Because the *moses* open-source toolkit (Koehn et al., 2007) supports both phrase-based as well as hierarchical machine translation, both integrated systems are available in *moses*.

8.1 Hierarchical Machine Translation for Domain Adaptation

We first show in which extent hierarchical machine translation can help domain adaptation in cases where phrase-based systems may lack of expressive power. We consider news as our first (or old) domain and medical as our second (or new) domain. For the same reasons as described in section 2, we work with the French-English language pair. Assume that the following French source sentence (FNews) and English reference (ENews) belong to the news domain.

- FNews : *Il a été retrouvé confiné dans une* ***enceinte***
- ENews : *He has been found hidden in a* ***building***

In this first sentence pair, the French noun *enceinte* is translated into *building* and no reordering is performed. Now consider the sentences FMed and EMed, which belong to the medical domain.

- FMed : *medicament pour personnes diabétique* ***enceinte***
- EMed : *medication for* ***pregnant*** *diabetic persons*

In this second sentence pair, the French adjective *enceinte* is translated into *pregnant* and is moved in front of the segment *diabetic persons*. In order to obtain a correct translation and reordering of *enceinte* in both domains above, we need a model that tells us that in the sequence *personne diabétique* ***enceinte***, the word *enceinte* has to be translated as *pregnant* and moved in front of the sequence. In order to obtain this information using a phrase-based system, a phrase-pair like PMed below has to be seen in training.

- PMed : personne diabétique **enceinte** → **pregnant** diabetic person
- PNews : **dans une enceinte** → **in a building**

In the same fashion, a phrase-pair like PNews has to be seen in order to correctly translate and reorder sentences FNews and FMed. Otherwise, there is no direct way to assign high probabilities to such sequences and the reordering decision is deferred to the lexical reordering model. Note that both phrases are relatively domain specific and hence not likely to bee both seen during training.

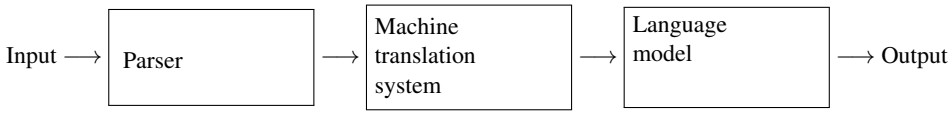
Within a hierarchical, or in general a syntax-based framework, the correct translation and reordering of sentences FNews and FMed only requires the following rules getting a high score when applied within the right domain :

- RNews : X **enceinte** → X **building**
- RMed : X **enceinte** → **pregnant** X
- RMedS : NP **enceinte** → **pregnant** NP

It is obvious that such rules are more likely to be extracted from the training set than complete phrases such as PMed.

8.2 Syntax Based SMT

Unlike phrase-based SMT systems, where phrasal segmentation is performed on sentences provided to the machine translation system, syntax-based SMT systems decode by parsing the provided input. Note that by "syntax-based", we denote all MT systems using parsing for decoding. These include, among others, hierarchical, tree-to-string and string-to-tree as well as tree-to-tree systems. The figure below displays a syntax-based decoding pipeline.



When training a syntax-based system, syntactic labels obtained from parse trees can be used to annotate non-terminals in the translation model. The annotation can either be directly attached to the SCFG rules such as in rule RMedS in the previous section. In this case, syntactic constituents have to be directly matched during decoding. This approach is often referred to as "hard syntax". Another possibility consists in adding linguistic constraints to hierarchical models using feature functions. This approach is often referred to as "soft syntax". In general, models using hard syntactic constraints tend to have coverage problems as noted by Ambati & Lavie (2008). However, work has been done to improve coverage such as inexact constituent matching (Zollmann & Venugopal, 2006), joint decoding (Liu et al., 2009) or parse relaxation (Hoang & Koehn, 2010).

8.2.1 Hard Syntactic Constraints for Domain Adaptation

Using hard syntax for domain adaptation has several drawbacks mainly related to the coverage problems encountered using this approach. For instance, while a sentence like FMed fits well in such a system, translation of FNews is more difficult. By parsing sentence FMed and applying the SCFG rules SC1 to SC4 below, the segment *personne diabétique enceinte* can be correctly translated and reordered. A the top of the derivation, a rule like SC1 can be picked because *personne diabétique enceinte* is very likely to be labeled as an NP by a French parser. Furthermore, SC1, as well as SC2 to SC4, have a sufficient level of generality to be likely to be seen during training.

- SC1 : SENT/SENT $\rightarrow$ <NP enceinte , pregnant NP>
- SC2 : NP/NP $\rightarrow$ <NN ADJ , ADJ NN>
- SC3 : NN $\rightarrow$ <personne , person>
- SC4 : ADJ $\rightarrow$ <diabétique , diabetic>

However, problems arise when trying to translate sentences like FNews with rules having *enceinte* as lexical item because the part of the input sentence covered by the non-terminal in the rule is no complete syntactic constituent. In other words, in the segment *confiné dans une enceinte*, the segment *confiné dans une* does not compose a complete syntactic constituent. In this case, hard syntactic constraints force the application of a rule like SENT/SENT $\rightarrow$ <VPART PREP DET enceinte , VPART PREP DET building>. The forced application of such rules causes a loss of generality over rules like X/X $\rightarrow$ <X enceinte , X building>. SCFG rules containing linguistic syntactic constituents are, first, less likely to be seen in training and, second, tend to apply badly on unseen data.

8.2.2 Soft Syntactic Constraints for Domain Adaptation

Using a hierarchical Phrase-based SMT system instead of a system using (hard) syntactic labels allows us to avoid restrictions related to non-matching constituents. For instance, in a hierarchical system, the word *enceinte* in sentence FNews above can be translated by using rule RNews (X **enceinte** $\rightarrow$ X **building**). However, the removal of syntactic labels from SCFG rules highly increases structural ambiguity. Rules such as RNews can indeed by applied to any French sentence containing the word *enceinte* with X having any possible span withdraw. Combining syntactic **features** with a hierarchical system restricts the structural ambiguity of hierarchical rules while allowing rule application across syntactic constituents. This has been shown, among others by Marton & Resnik (2008), Chiang (2010) and Simianer et al. (2012). As an example, consider the following segments in FNews and FMed after parsing :

- ($\boxed{\text{confiné}_{VPART}\ \text{dans}_{PREP}\ \text{une}_{DET}}$ $\text{enceinte}_{NN})_{SENT}$

- ($\boxed{\text{(personne}_{NN}\ \text{diabétique}_{ADJ})_{NP}}$ $\text{enceinte}_{ADJ})_{NP}$

When translating the input sentence FNews from the News domain, the information that the word *enceinte* is a noun (NN) and that its parent constituent is the whole sentence (SENT) helps the system to correctly chose rule RNews (X **enceinte** → X **building**) in a derivation. In the same fashion, when translating sentence FMed, the information that *enceinte* is an adjective (ADJ) and is located in a noun phrase (NP) indicates that the rule RMed (**enceinte** → **pregnant** NP) should be picked. Hence, for domain adaptation, soft syntax allows the system to work with rule having a high expressive power while decreasing structural ambiguity with a feature encouraging constituent matches.

8.3 Integration of VW in a Hierarchical SMT System

8.3.1 Estimation of a Syntax Feature Score

The integration of VW in hierarchical moses allows, first, to integrate soft syntactic features in this system. As seen in section 8.2.2, such features help to reduce the structural ambiguity inherent to a hierarchical system without loss of generality. This in turn allows better adaptation to new domains. VW is trained on a large word-aligned parallel corpus parsed on the source language (French) side. The classifier is then called during decoding and the obtained predictions define a syntactic score which can be used as one feature in the log linear model. A large number of features can potentially be used to train VW. Currently the following are used :

- Constituent and parent of applied rule
- Span width of applied rule
- Type of reordering (multiple non-terminals)

In the long run, we plan to integrate more syntactic features in the system using not only source but also target context information.

8.3.2 Estimation of a PSD probability

Because the integration of VW in a hierarchical system allows to handle a large number of features, a PSD model 5 can be added to syntactic features. The integration of PSD features further reduces structural ambiguity by helping the system to chose rules containing the correct lexical items. For instance, in sentence FNews and FMed, given again below, knowing that the token "confiné" occurs in FNews helps to select R3 (X **enceinte** → X **building**) while knowing that "personne" occurs in FMed helps to select rule RMed (X **enceinte** → **pregnant** X).

- $\boxed{\textbf{personne}\ \text{diabétique}}$ enceinte ⇒ pregnant $\boxed{\text{diabetic patient}}$
- $\boxed{\textbf{confiné dans}\ \text{une}}$ enceinte ⇒ $\boxed{\text{hidden in a}}$ building

Hence, all features composing the PSD model integrated into phrase-based moses are also integrated into the hierarchical version. In this setup, the combination of syntax and psd scores define one feature in the log-linear model. Among others, the following PSD features have been integrated in hierarchical moses :

- French (source) context of rule
- Source and Target of rule
- Bag of words inside of rule
- Bag of words outside of rule
- Aligned terminals
- Rule scores (e.g. p(e|f))

8.3.3 Calls to VW during decoding

In a hierarchical system, the number of translation options provided for a given segment of the input sentence is typically larger than in a phrase-based system. More precisely, for each matched segment (e.g. patiente diabétique enceinte), we have :

- N possible rule source sides :
  - X/X $\rightarrow$ <X enceinte , ...>
  - X/X $\rightarrow$ <patiente X , ...>
  - X/X $\rightarrow$ <patiente X enceinte , ...>

Then we have, **for each source side** :

- M possible target sides :
  - X/X $\rightarrow$ <X enceinte , pregnant X>
  - X/X $\rightarrow$ <X enceinte , X building>

A phrase-based system only matches one source phrase (with M corresponding targets) to each considered segment. Because during decoding VW is called for each translation option, that is each matched rule, the number of calls to VW potentially becomes very high within a hierarchical system. However, the runtime slowdown is manageable. In order to measure this, we trained a hierarchical system using 29515 sentence-pairs from the medical domain. This training set has also been used to train VW. Then we decoded 2000 sentences using a standard hierarchical system and a system integrating VW. The first system uses 7 minutes to decode the given input while the second system uses 21 minutes. Note that the training set is very small and hence results in very small rule tables. Further note that the slowdown is reduced by the setting of an upper limit to the number of rules used to build translation options.

9 Domain Adaptation for PSD

In this section, we apply a number of domain adaptation techniques on the PSD data in order to train a model that uses both old and new domain data and outperforms the individual models. In Section 9.6, we intrinsically evaluate different domain-adaptation techniques and compare them to the baselines on the EMEA32 and Science domains.

9.1 Baselines

The natural baseline for domain adaptation is to concatenate the old and new data and train a classifier on it. The result of this concatenation could be different depending on the relative size of old and new and their dissimilarity. If old is different from new and is much larger, concatenating them would probably hurt the performance.

Figure 8 illustrates the PSD accuracy for various baselines on EMEA32 and Science. The first and second bars of each group represent EMEA32 and Science results respectively. *Old* and *New* refer to old-only and new-only models. *Old + New* shows the accuracy of the classifier built on the concatenation of the old and new data. The next baseline is the classifier that picks the most frequent translation of source phrases ($\text{argmax}\, p(e|f)$). The frequency statistics for this baseline are collected from the phrase-table built by concatenating of the old and new phrase-tables. Finally, the last baseline shows the percentage of time a random guessing would pick the correct translations. As the figure shows, the concatenation baselines are slightly worse than the new-only models in both domains.

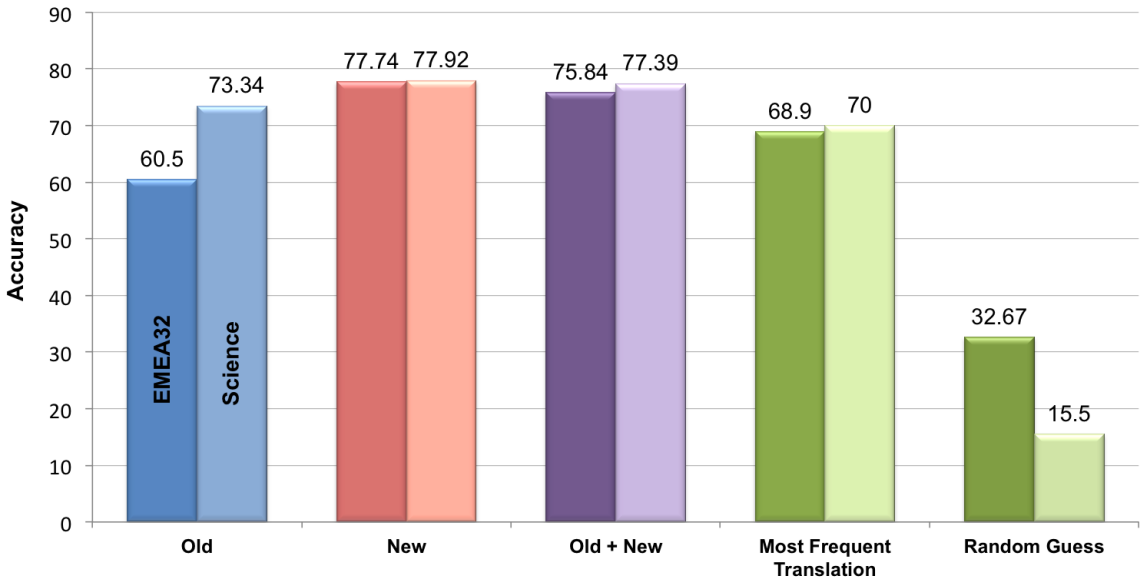


Figure 8: The PSD accuracy for different baselines on EMEA32 and Science. The first and second bars of each group represent EMEA32 and Science results respectively.

9.2 Frustratingly Easy DA

The *frustratingly easy domain adaption* technique by Daumé III (2007) distinguishes between features that are common between the old and new domains and those that have different interpretations across domains. This method augments features in the old-domain training data by making a copy of them. Using this techniques, features that have different meanings in old from new do not get canceled out when combining the two training sets. Daumé III (2007) shows that this technique is very effective in domain adaptation while it requires only a few lines of code to implement.

9.3 Instance Weighting

Concatenating old and new domain data treat old and new instances in the same manner. This can degrade the scores especially when old is large and very different from new. A better approach for domain adaptation would be to use only

instances from old that are similar to new. Instead of setting a threshold and classifying the old-domain instances into two classes (i.e. similar and dissimilar), we can weight each instance in the old-domain data based on the probability with which that instance belongs to new. In other words, the more similar an old-domain instance is to the new domain, more important it gets for the PSD classifier.

The first step is to learn a domain separator classifier. This is done by striping phrase-identity features from source and target and learn a VW classifier to distinguish between old (with label -1) from new (with label +1). Since there are many more features than instances[5], we need to regularize the training step. Otherwise, the domain separator model overfits (with domain-separation accuracy of 99%) and cannot generalize well.Once the domain separator model is learned, it is applied to the old-domain data to classify the instances into two classes. However, we use only the classification probability for each instance and we use it to weight them in old. The weighted old data, then, gets concatenated to the original new data and a new classifier is trained on the concatenation.

Using this technique, we allow the classifier not to get biased heavily towards the old-data instances which are larger than new-data instances. Table 13 shows the effect of the regularizer parameter value (λ) on the domain-separator classifier as well as on the final classifier. Based on the results illustrated in this table, there is a very weak relationship between the accuracy of the domain-separator classifier and that of the final classifier.

L1 λ	**Old Precision**	**Science Precision**	**Domsep Accuracy**	**Classifier Accuracy**
1e-03	91.21	67.81	82.31	77.93
1e-04	92.71	79.97	87.86	77.97
1e-05	93.85	84.8	90.41	78.01
1e-06	95.67	90.04	93.52	78.02

Table 13: The effect of different L1 parameter values on the domain separator classifier and the final classifier.

9.4 New + Old Prediction Feature

This is a simple adaptation technique where we do not fully use the old-domain model/data. Instead, we only use the predictions of the old model as features in the new-domain data. Particularly, we train a model on the old-domain data and apply it to the new-domain data (and the dev-set). Then, for each instance in the new-domain data, we add an indicator feature on the predicted label (i.e. argmax). Alternatively, for every label, we can add a numeric feature indicating the level to which the old-only model is confident about that label being the correct one. Our results show that using the full old-model score (i.e. the latter case) slightly improves the accuracy of the classifier. When we use only indicator features on one of the labels in each group, the accuracy is 77.88% while using the full old-model score on all labels, we get 78.04% accuracy.

9.5 Model interpolation

In model interpolation, two separate models are trained on old and new and these models are interpolated linearly or log-linearly.

$$P_{\text{linear}}(e|f) \quad \propto \quad \lambda_1 P_{old}(e|f) + \lambda_2 P_{new}(e|f)$$

$$P_{\text{log-linear}}(e|f) \quad \propto \quad P_{old}(e|f)^{\lambda_1} P_{new}(e|f)^{\lambda_2}$$

However, for our experiments since the accuracy of the old-only model is significantly lower than that of the new-only model (in both domains), log-linear interpolation would hurt the accuracy. The interpolation weights are learned using cross-validation on the dev-set. The model interpolation can be done offline or the predictions can be mixed online.

[5]The number of total features is two orders of magnitude larger than the number of training instances due to using quadratic features

9.6 Adaptation Results

Table 14 shows the accuracies for different baselines and different domain adaptation techniques that were used. The results are based on two subsets of the EMEA training-set: EMEA16 and EMEA32. *Unseen Dev* refers to a subset of *Dev* that does not have an overlap with the training-set. Similarly, Table 15 shows the results on Science unseen dev-set.

System	Dev		Unseen Dev	
	EMEA16	EMEA32	EMEA16	EMEA32
Old	61.5		55.84	
New	78.85	77.74	73.04	72.38
Old + New	76.85	75.84	70.57	69.66
FEDA (Old Aug)	78.26	77.33	72.60	71.76
Instance Weighting	78.12	77.48	72.89	71.49
Old Initialized New	75.35	73.49	66.47	65.09
New + Old Prediction Feature	78.75	78.09	72.86	72.90
Linear Mixture	78.99	77.99	73.39	72.91

Table 14: Unadapted and adapted PSD classifier accuracy on EMEA16 and EMEA32

System	Dev(unseen)
Old	73.34
Science	77.92
Old + Science	77.39
FEDA (Old only)	77.92
Instance Weighting	77.97
New + Old Prediction Feature	77.88
New + Old Prediction Feature (full score)	78.04

Table 15: Unadapted and adapted PSD classifier accuracy on Science

As Table 14 and 15 show, the domain adaptation techniques we used over-perform our baselines. However, the difference is not significant. Preliminary inspections revealed that among the dev-set instances that new-only model was wrong in classification, only about 4% are correctly classified by the old-only model (for both EMEA and Science). In other words, the old-only model has little information to add to what the new-only model knows already and this is also consistent with the results we showed in Section 9.1 where the accuracy of the concatenation baseline falls behind the new-only model in both models. We suspect this is due to the large and noisy old-domain data and the fact that the domains are very apart. We need to experiment with more sophisticated domain adaptation approaches. The following two contingency tables report new-only and old-only inter-model agreements on the dev-set instances.

Figure 9 illustrates the learning curve for different DA techniques. We ran all the experiments for 10 iterations and recorded the accuracy of the intermediate models. The accuracies of all models go down after a couple of iterations (the old-only model gets worse after the first iteration). The exception is the instance-weighting model that performs almost constantly starting from the 5th iteration. This figure suggests that the models are over-fitting and we need to apply some regularization penalty or train them for fewer iterations.

Old ↓ EMEA →	Correct	Incorrect
Correct	57%	4%
Incorrect	21%	18%

Old ↓ Science →	Correct	Incorrect
Correct	63%	4%
Incorrect	13%	20%

Table 16: Old-only and new-only inter-model agreements on the dev-set instances.

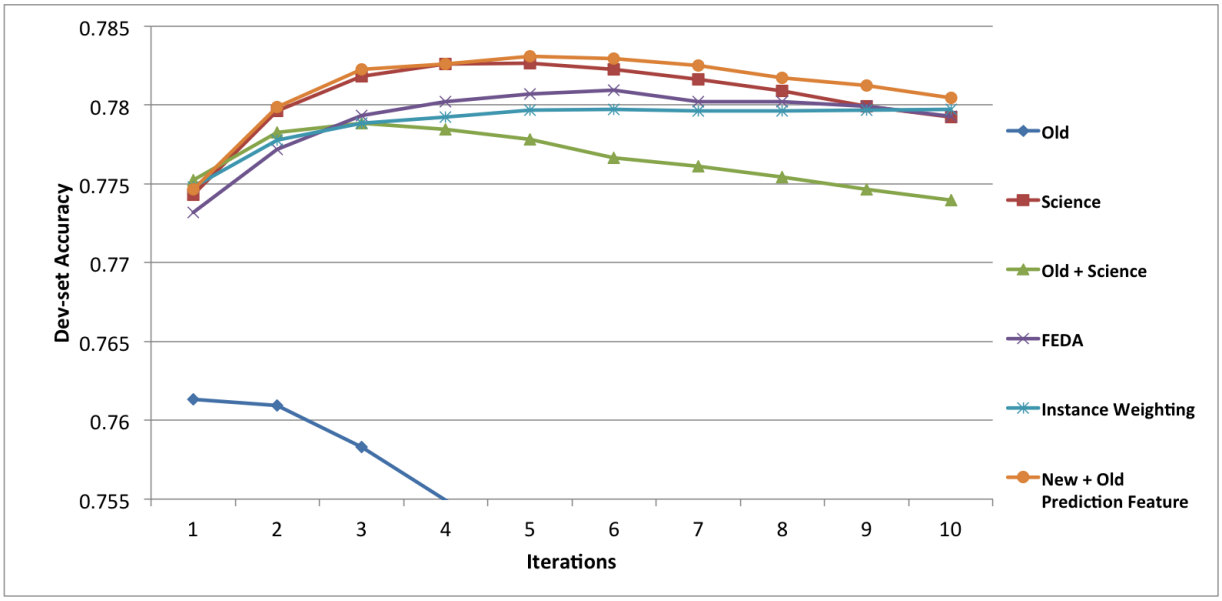


Figure 9: Learning curve for different baselines and DA approaches over 10 iterations on the Science domain.

10 Introduction to Vocabulary Mining

Many of the prior sections have addressed the situation where there is a large amount of parallel data in the new domain. Such parallel data often makes significant inroads in the problems of "seen" and "sense", that is, new domain source language words that are either unseen in the old domain or that require new target language translations. In the absence of parallel data, however, these are likely the most crucial problems.

The next three sections address several important questions in this area. First, we discuss one method for extracting new translation pairs from comparable data. This could be used to provide translations for unseen words, and also to augment the translation options for words that were already seen.

Since we may not want to mine translations for all possible words, and we might not want to augment the translation options for all words, we next focus on a method for detecting when a word may require a new translation. Detecting the need for a new sense is cast as a problem in classification.

There may also be a number of clustered sub-domains nestled inside our so-called OLD domain. Especially when the provenance of the data is broad (e.g. within scientific abstracts, or in data gathered from the web) and the scale of data begins to grow, latent sub-domains or topics seem likely to occur. We build upon prior work exploiting latent topics, describing new ways to incorporate latent topic information as features in our system and new models for finding latent topics.

Finally, we explore several methods for exploiting low-dimensional continuous representations of words for translation mining. Beginning from a distributional representation of words, we can learn projections into low-dimensional spaces that minimize distance between translation pairs. Prior work is primarily concerned with learning at the type level; here we extend that work to consider learning at the token level.

11 Marginals Technique for Extracting Word Translation Pairs

As we showed in Sections 3 and 4, a major challenge when using a machine translation model trained on OLD-domain parallel data (e.g. parliamentary proceedings) to translate NEW-domain text (e.g. scientific articles) is the large number of out-of-vocabulary (OOV) and new-translation-sense (NTS) words. Acquiring translations for such words is particularly critical in the case of little or no NEW-domain parallel data. In this section, we present a method to identify new translations of both known and OOV source words that uses only comparable document pairs in the NEW-domain. Starting from a joint distribution of source-target word pairs derived from the OLD-domain parallel corpus, our method recovers a new joint distribution that matches the marginal distributions of the NEW-domain comparable document pairs, while minimizing the divergence from the OLD-domain distribution. We also incorporate useful orthogonal sources of information based on string similarity and monolingual word frequencies. Adding these learned translations to our French-English MT model results in gains of over 2 BLEU points over strong baselines.

11.1 Overview of Marginals Technique

In this work, we seek to learn a joint distribution of translation probabilities over all source and target word pairs in the NEW-domain. We begin with a maximum likelihood estimate of the joint based on a word aligned OLD-domain corpus and update the joint using NEW-domain comparable data. We define a model based on a single comparable corpus and then modify it slightly to learn from any number of comparable *document pairs*, or document aligned comparable corpora. After learning a new joint distribution over all word pairs, we use it to update our SMT model. This approach allows us to learn translations for previously OOV words (e.g. French *cisaillement* and *perçage*, which translate as *shear* and *drilling*, in the scientific domain) as well as new translations for previously observed NTS words (e.g. *enceinte* translates as *enclosures*, not *place*, in the scientific domain).

Our approach depends crucially on finding comparable document pairs relevant to the NEW-domain. Such pairs could be derived from any number of possible sources, and documents may be linked based on timestamps (e.g. news articles) or topics (inferred or manually labeled). We use Wikipedia[6] as a source of comparable pairs. So-called "interwiki links" (which link Wikipedia articles on the same topic written in different languages, such as English and French) act as rough guidance that pages may contain very similar information. Our approach does not exploit any Wikipedia structure beyond this initial signal, and thus is portable to alternate sources of comparable articles, such as multilingual news articles covering the same event.

11.2 Previous Work

There is a plethora of prior work on learning bilingual lexicons from monolingual and comparable corpora. Approaches have used a variety of signals including distributional, temporal, and topic similarity (Rapp, 1995; Fung & Yee, 1998; Rapp, 1999; Schafer & Yarowsky, 2002; Schafer, 2006; Klementiev & Roth, 2006; Koehn & Knight, 2002; Haghighi et al., 2008; Mimno et al., 2009; Mausam et al., 2010). Prochasson & Fung (2011) extract translations for rare medical terms. However, all of this prior work stops short of applying bilingual lexicons to end-to-end MT. In this work, we supplement a baseline MT system with learned translations.

Our approach bears some similarity to those of Ravi & Knight (2011), Dou & Knight (2012), and Nuhn et al. (2012) in that we hope to learn a translation distribution despite a lack of parallel data. However, we focus on the domain adaptation setting: parallel data in some OLD-domain acts as a starting point (or prior) for this translation distribution. In fact, we believe that even in low resource settings, it is reasonable to assume that some initial bilingual dictionary can be obtained, for example through crowdsourcing (Callison-Burch & Dredze, 2010) or pivoting through related languages (Schafer & Yarowsky, 2002; Nakov & Ng, 2009).

Daumé III & Jagarlamudi (2011) mine translations for high frequency OOV words in NEW-domain text for the purpose of domain adaptation. Although that work shows significant MT improvements, it is based upon distributional similarity, thus making it difficult to learn translations for low frequency source words with sparse word context counts. Our model allows us to incorporate any number of signals from monolingual corpora, including distributional similarity. Additionally, this important prior work reports results based on artificially created monolingual corpora taken from separate source and target halves of a NEW-domain parallel corpus, which may have more lexical overlap with the corresponding test set than we could expect from true monolingual corpora. Our work mines NEW-domain-like document pairs from Wikipedia. In the results below, we directly compare supplementing a baseline SMT model with

[6] www.wikipedia.org

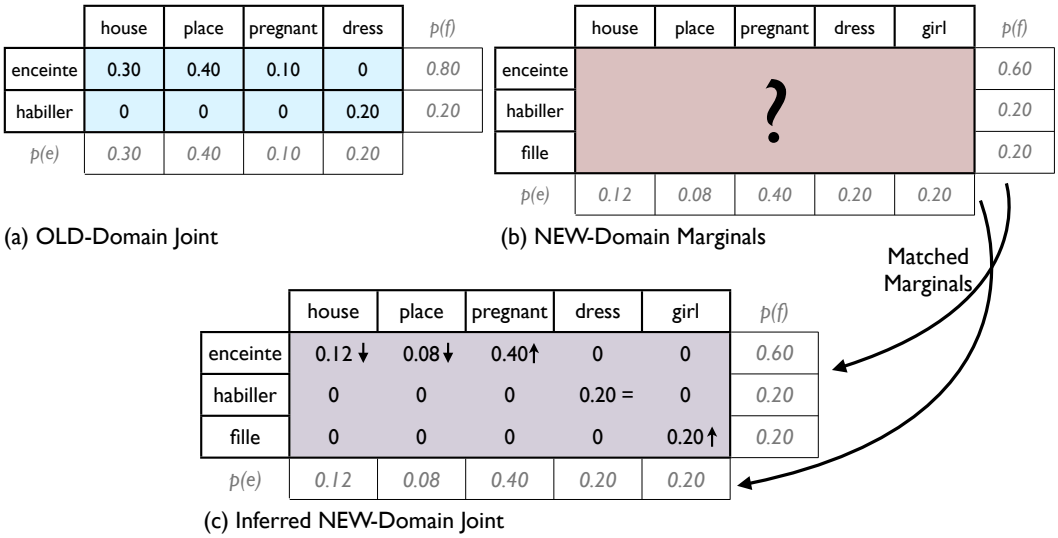


Figure 10: Example of starting with a joint distribution derived from OLD-domain data and inferring a NEW-domain joint distribution based on the intuition that the new joint should match the marginals that we observe in NEW-domain comparable corpora. In this example, a translation is learned for the previously OOV word *fill*, and *pregnant* becomes a preferred translation for *enceinte*.

the translations that our model learns and those learned by the model described in Daumé III & Jagarlamudi (2011), keeping data resources constant.

11.3 Model

Our goal is to recover a probabilistic translation dictionary in a NEW-domain, represented as a joint probability distribution $p^{(\text{new})}(s,t)$ over source/target word pairs. At our disposal, we have access to a joint distribution $p^{(\text{old})}(s,t)$ from the OLD-domain (computed from word alignments), plus comparable documents in the NEW-domain. From these comparable documents, we can extract raw word frequencies on both the source and target side, represented as marginal distributions $q(s)$ and $q(t)$. The key idea is to estimate this NEW-domain joint distribution to be as similar to the OLD-domain distribution, subject to the constraint that its marginals match those of q.

To illustrate our goal, let us consider an example. Imagine in the OLD-domain parallel data we find that *accorder* translates as *grant* 10 times, and as *tune* 1 time. In the NEW-domain comparable data, we find that *accorder* occurs 5 times, but *grant* occurs only once, and *tune* occurs 4 times. This clearly demonstrates that *accorder* no longer translates as *grant* most of the time; perhaps we should shift much of its mass onto the translation *tune* instead. Figure 10 shows the intuition.

First we present an objective function and set of constraints over joint distributions to minimize the divergence from the OLD-domain distribution while matching both the source and target NEW-domain marginal distributions. Next we explore several extensions to augment this objective function and capture additional information beyond the marginals. Optimizing this objective with a single pair of source and target marginals can be performed using an off-the-shelf solver. In practice, though, we have a large set of document pairs, each of which can induce a pair of marginals. Using these per-document marginals provides additional information to the learning function but would overwhelm a common solver. Therefore, we present a sequential learning method for approximately matching the large set of document pair marginal distributions. Finally we discuss a method for obtaining comparable document pairs relevant to the NEW domain.

11.4 Marginal Matching Objective

Given word-aligned parallel data in the OLD-domain and source and target comparable corpora in the NEW-domain, we first estimate a joint distribution $p^{(\text{old})}(s,t)$ over word pairs (s,t) in the old domain, where s and t range over source and target language words, respectively. We use a simple maximum likelihood estimate based on non-null automatic word alignments (using grow-diag-final GIZA++ alignments (Och & Ney, 2003)). Next, we estimate source and target marginal distributions, $q(s)$ and $q(t)$, using simple relative frequency counts over the source and target comparable

corpora. Our goal is to recover a joint distribution $p^{(\text{new})}(s,t)$ for the new domain that matches the marginals, $q(s)$ and $q(t)$, but is minimally different from the original joint distribution, $p^{(\text{old})}(s,t)$.

We phrase this as a linear programming problem[7]:

$$p^{(\text{new})} = \arg\min_{p} \; \mid p - p^{(\text{old})} \mid_1 \tag{1}$$
$$\text{subject to: } \sum_{s,t} p(s,t) = 1, \quad p(s,t) \geq 0$$
$$\sum_{s} p(s,t) = q(t), \quad \sum_{t} p(s,t) = q(s)$$

Here, joint probability matrices p and $p^{(\text{old})}$ are interpreted as large vectors over all word pairs (s,t). This minimization is subject to the normal sum of probabilities and nonnegative probabilities constraints (the first two constraints), as well as our novel marginal matching constraints (the final two constraints).

In addition to forcing the difference between the old and new matrices to be sparse, we would also like the new matrix to remain as sparse as possible, following prior work (Ravi & Knight, 2011). That is, we believe that the model should add as few translation pairs as possible to account for the changes in the marginal distribution. We add a regularization term to capture this intuition:

$$\Omega(p) = \sum_{\substack{s,t: \\ p^{(\text{old})}(s,t)=0}} \lambda_r \times p(s,t) \tag{2}$$

If the old domain joint probability $p^{(\text{old})}(s,t)$ was nonzero, there is no penalty. Otherwise, the penalty is λ_r times the new joint probability $p(s,t)$. To encourage the removal of translation pairs that become unnecessary in the new domain, we use a λ_r weight on this regularization term that is greater than one. Doing so makes the benefit of a more sparse matrix overwhelm the desire for preventing change. Any value greater than one appears to suffice; we use $\lambda_r = 1.1$ in our experiments.

Inspired by the term that encodes a preference for sparse matrices, $\Omega(p)$, we include additional orthogonal cues that words are translations of one another in the objective function (Eq (1)) with additional terms, $f_j(p)$:

$$p^{(\text{new})} = \arg\min_{p} \; \mid p - p^{(\text{old})} \mid_1 + \Omega(p) + \sum_j \lambda_f f_j(p)$$

We define three additional $f_j(p)$ terms:

Penalty for word frequency differences: Most of the time, rare words should align to rare words, common words should align to common words, and rare words should not align to common words. The penalty is zero if the monolingual frequency of t and s is exactly the same and one if either t or s has a frequency of zero. The penalty is close to zero if the relative frequency difference is small and close to one if it is large.

$$f_1(p) = p(s,t) \cdot \frac{|\text{freq}(t) - \text{freq}(s)|}{\text{freq}(t) + \text{freq}(s)}$$

Penalty for edit distance: Words that are spelled similarly are often translations of one another. Here, if the normalized Levenshtein edit distance between s *without accents* and t is less than 0.2, no penalty is applied and a penalty of 1 is applied otherwise. We chose the 0.2 threshold manually.

$$f_2(p) = p(s,t) \cdot \begin{cases} 0 & \text{if } \frac{\text{lev}(t,\text{strip}(s))}{\text{len}(s)+\text{len}(t)} < 0.2 \\ 1 & \text{otherwise} \end{cases}$$

[7] Initially we experimented with a quadratic penalty on divergence, but here a sparse set of differences seemed to produce better results on a small dataset.

Penalty for differences in document-pair co-occurrence: Writing $D(w)$ for the vector indicating the document pairs in which w occurs, if $D(s)$ and $D(t)$ are dissimilar, it is less likely (s,t) is a valid translation pair. We weighted $D(w)$ entries with BM25 (Robertson et al., 1994). We use the set of 50,000 document-pairs which are most *NEW-domain-like*, to compute these vectors.

$$f_3(p) = p(s,t) \cdot (1 - \cos(D(s), D(t)))$$

After manual tuning on a small amount of data, we set $\lambda_f = \frac{1}{3}$.

The objective can be optimized by any standard LP solver; we use the Gurobi package (Gurobi Optimization Inc., 2013).

11.5 Document Pair Modification

While the above formulation can work for any setting in which we have access to comparable *corpora*, in many cases we actually have access to comparable *documents*: for instance, those given by inter-language links on Wikipedia. We modify our objective slightly because we would like to take advantage of document pair alignments. That is, since we have information about document correspondence within our comparable corpus, we would like to match the marginals for *all document pairs*.

An initial formulation our problem with multiple comparable document pairs might require the $p^{(\text{new})}$ marginals to match *all* of the document marginals. In general, this constraint set will be empty. Instead, we take an incremental, online solution. Specifically, we consider a single comparable document pair at a time. For each pair, we solve the optimization problem in Eq (1) to find the joint distribution minimally different from $p^{(\text{old})}$, while matching the marginals of *this pair*. Again, we use the Gurobi package to optimize the objective, now for each document pair. This gives a new marginal distribution, tuned specifically for this pair. We then update our current guess of the new domain marginals *toward* this document-pair-specific distribution, much like a step in stochastic gradient ascent.

More formally, suppose that before processing the kth document we have a guess at the NEW-domain joint distribution, $p^{(\text{new})}_{1:k-1}$ (the subscript indicates that it includes *all* document pairs up to and including document $k-1$). We first solve Eq (1) just on the basis of this document pair, to get a joint distribution $p^{(\text{new})}_k$, which depends *only* on the kth document pair. Finally, we form a new estimate of the joint distribution by moving $p^{(\text{new})}_{1:k-1}$ in the direction of $p^{(\text{new})}_k$, via:

$$p^{(\text{new})}_{1:k} = p^{(\text{new})}_{1:k-1} + \eta_u \left[p^{(\text{new})}_k - p^{(\text{new})}_{1:k-1} \right]$$

Here, η_u is a learning rate parameter, set to 0.001 in our experiments[8].

In order to account for the number of identical or near identical French to English translations, we make one additional modification to this algorithm. After each iteration of learning, we give a slight boost to $p^{(\text{new})}_k(s,t)$ for each word s and its identity, stripped of all accents, strip(t). The size of the update is based on the frequencies of s and t. In particular, we set pair-specific learning rates to be $\lambda_{s,t} = \lambda_u \cdot .00001 \cdot max(1, min(\text{freq}(s), \text{freq}(t))$, where λ_u is the same as above. So, if both s and t are seen frequently in our entire monolingual corpus, the increase is relatively large. If t is never seen, the increase is relatively small. After minimally artificially increasing identity translations, we normalize the learned joint distribution. Note that in the results presented below, there is often a relatively large jump in performance from the beginning of learning to the first reported step. This is a result of this artificial identity-translation probability boosting. Although this artificial boosting helps performance, our marginal matching-based method for learning translations goes far beyond identical and near-identical translations.

Our updates are, in a sense, like stochastic gradient descent, using an empirical estimate of the gradient rather than an analytic one. Unlike other empirical gradient estimates such as finite differences (FD) (Berman et al., 1987; Blum, 1954) and simultaneous perturbation (SP) (Spall, 1992), the estimate is based on the difference between the current point and an optimal point for the specified subproblem. Such a value is expensive to compute but is likely to give an informative search direction. In order to accelerate learning, we parallelize our algorithm. We have 8 parallel learners update an initial joint distribution based on 100 document pairs and merge results using an average over the 8 learned joint distributions.

[8] We manually tuned this parameter, based on intrinsic results over a very small corpus.

11.6 Comparable Data Selection

It remains to select these comparable document pairs. We assume that we have enough monolingual new-domain data in one language to rank comparable document pairs (here, Wikipedia pages) according to how *NEW-domain-like* they are. In particular, we estimate the similarity to a source language (here, French) corpus in the new domain. For our experiments, we use the French side of a new-domain parallel corpus[9]. We could have also targeted our learning even more by using our NEW-domain development and test sets themselves. Doing so would increase the chances that our source language words of interest appear in the comparable corpus. However, to avoid overfitting any particular test set, we use the larger French side of the training data.

For each Wikipedia document pair, we measure the percent of French unigram, bigram, trigram, and 4-gram types that are observed in the French monolingual new domain corpus and rank document pairs by the geometric mean between the four overlap measures. In the results below, we show learning curves over these ranked document pairs. As mentioned, we also use this ranked list of document pairs to calculate the document-pair co-occurrence penalty. We did not explore using more sophisticated ways to identify NEW-domain-like Wikipedia pages, such as Moore & Lewis (2010), and it is possible that a better ranking algorithm would yield additional performance gains. However, qualitatively, the ranked of Wikipedia pages seemed very reasonable to the authors.

11.7 Experimental setup

11.7.1 Data

We use the French-English Hansard parliamentary proceedings[10] as our OLD-domain parallel corpus. Containing over 8 million lines of parallel training text, it is one of the largest freely available parallel corpora for any language pair. In order to simulate more typical data settings, we sample every 32nd line and use the resulting parallel corpus with $253,387$ lines and $5,051,016$ tokens to train our baseline model.

We test our model using three NEW-domain corpora: (1) the EMEA medical corpus (Tiedemann, 2009), (2) a corpus of scientific abstracts (anonymous, 2012), and (3) a corpus of translated movie subtitles (Tiedemann, 2009). We use development and test sets to tune our MT model and then evaluate end-to-end MT performance. We use the NEW-domain parallel training corpus *only* for language modeling, identifying NEW-domain-like comparable documents, and intrinsic lexicon induction evaluation. In all parallel corpora, we normalize English data for American spelling.

11.7.2 Machine translation

We use the Moses MT framework (Koehn et al., 2007) to build a standard statistical phrase-based MT model using our OLD-domain training data. Using Moses, we extract a phrase table with a phrase limit of five words and estimate the standard set of five feature functions (phrase and lexical translation probabilities in each direction and a constant phrase penalty feature). We also use a standard lexicalized reordering model and two language models based on the English side of the Hansard training data and the English side of the given NEW-domain training corpus. Features are combined using a loglinear model optimized for BLEU, using the n-best batch MIRA algorithm (Cherry & Foster, 2012). In our experiments below, we add new phrase pairs and new feature scores to the baseline phrase tables.

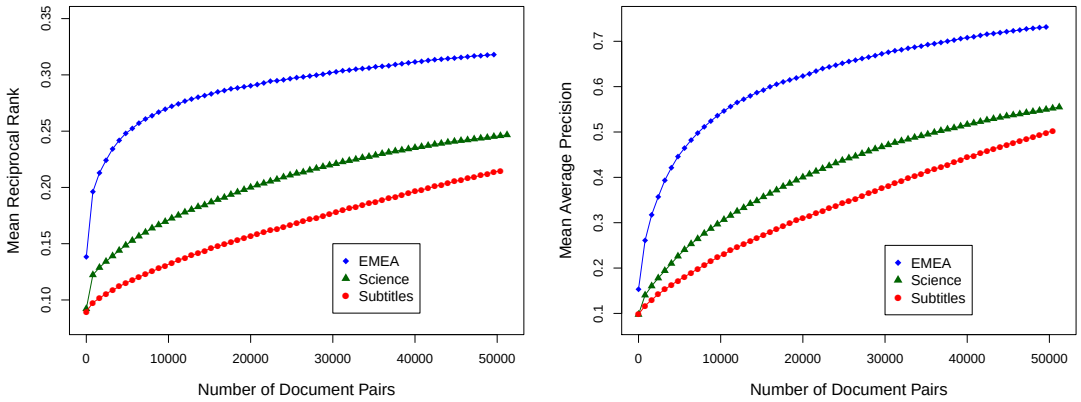
11.7.3 Experiments

For each domain, we use the marginal matching method described in Section 11.3 to learn a new, domain-adapted joint distribution, $p_k^{(\text{new})}(s,t)$, over all French and English words. We use the learned joint to compute conditional probabilities, $p_k^{(\text{new})}(t|s)$, for each French word s and rank English translations t accordingly. Before performing end-to-end MT experiments, we evaluate the learned joint directly, by comparing it to each joint distribution based on the word-aligned NEW-domain parallel corpora. We measure intrinsic performance using several metrics to compare the distributions and show learning curves over increasing numbers of comparable document pairs. In each MT experiment, we use both $p_k^{(\text{new})}(t|s)$ and $p_k^{(\text{new})}(s|t)$ as new feature scores for the new translation pairs and use constant values for the old translation features on the new translation pairs. Our end-to-end MT experiments vary the following:

- We append the top-k translations for each OOV French word to the phrase table, varying k.

[9] We could have, analogously, used the *target language* (English) side of the parallel corpus and measure overlap with the English Wikipedia documents, or even used both.

[10] http://www.parl.gc.ca



(a) Mean Reciprocal Rank (b) Mean Average Precision

Figure 11: Intrinsic bilingual lexicon induction results

French	OLD top $p^{(\text{old})}(e\|f)$	NEW top $p^{(\text{gold})}(e\|f)$	MM-learned top $p_k^{(\text{new})}(e\|f)$
cisaillement	-	shear strength shearing	shear viscous newtonian
courbure	-	curvature bending curvatures	curvature curved manifold
linéaires	linear	linear nonlinear non-linear	linear linearly nonlinear
récepteur	receiver	receptor receiver y1	receptor receiver receptors
ajustement	adjustment juggling adjusted	adjustment fit fitting	adjustment juggling fits
champ	field jurisdiction scope	field magnetic near-field	field magnetic fields

Table 17: Hand-picked examples of Science-domain French words and their top English translations in the OLD-domain, NEW-domain, and distribution learned by marginal matching. The first two French words are OOVs. The next two are not OOV but only appeared four and one time, respectively, in the training data and only aligned to a single English word. The last two examples are French words which appeared frequently in the training data but for which the word's sense in the new domain shifts (NTS words).

- We append the top-1 translation for all French words appearing fewer than c times, varying c.
- We compare giving existing phrase pairs constant values for the new features with using the new joint distribution in combination with phrase-internal alignments to score $p_k^{(\text{new})}(t|s)$ and $p_k^{(\text{new})}(s|t)$ for existing phrase pairs.

We also perform oracle experiments in which we identify translations for French words in word-aligned development and test sets and append these translations to baseline phrase tables. By doing so, we measure the percent of possible BLEU score gain that we realize using our learned estimates.

11.8 Results

11.8.1 Intrinsic evaluation

Before using the translations that we learn through marginal matching to supplement an end-to-end MT model, we evaluate our learned joint distribution $p_k^{(\text{new})}(s,t)$ intrinsically, by comparing it to the joint distribution taken from a word aligned NEW-domain parallel training corpus, $p^{(\text{gold})}(s,t)$. Figure 11 shows the mean reciprocal rank and mean average precision of learned joint distributions, $p_k^{(\text{new})}(s,t)$, as a function of the number of comparable document pairs used in learning. We learn over the $50,000$ document pairs which are most similar to each NEW-domain. Although it appears we could make some minimal additional gains by learning over more than $50,000$, performance is fairly stable at that point.

We experimented with making multiple learning passes over the document pairs and observed relatively small gains from doing so. In all experiments, learning from some number of additional new document pairs resulted in higher intrinsic performance gains than passing over the same number of document pairs which were already observed.

In the case of OOV words, it's clear that learning something about how to translate a previously unobserved French word is beneficial. However, our learning method also learns domain-specific new-translation senses (NTS). Table 17 shows some examples of what the marginal matching method learns for different types of source words (OOVs, low frequency, and NTS).

	Science	EMEA	Subs
Simple Baseline	21.91	23.67	**13.18**
Accent-Stripped	22.20	24.45	13.13
Top-1 Edit Dist	22.10	24.35	12.95
Top-5 Edit Dist	21.09	22.71	12.54
Top-1 Doc Sim.	**22.43**	**25.03**	13.02
Top-5 Doc Sim.	22.06	24.42	12.90
Top-1 CCA Distrib.	21.41	-	-
Top-5 CCA Distrib.	20.90	-	-
Top-1 MM	**23.83**	**26.65**	13.03
Top-3 MM	22.63	25.14	12.97
Top-10 MM	22.22	23.89	12.96

Table 18: BLEU score results using several baseline phrase tables and phrase tables augmented with top-1, top-3, and top-10 marginal matching (MM) translations for each OOV French word.

	Science	EMEA	Subs
Strongest Baseline	22.43	25.03	**13.18**
Freq < 1 (OOVs)	23.83	26.65	13.03
Freq <11	**24.64**	26.88	13.06
Freq < 101	24.34	**27.14**	12.91

Table 19: BLEU scores comparing phrase tables augmented with top-1 translations for each French word with the indicated OLD training data frequencies.

11.8.2 MT evaluation

By default, the Moses decoder copies OOV words directly into its translated output. In some cases, this is correct (e.g. *ensembles*, *blumeria*, *google*). In other cases, French words can be translated into English correctly by simply stripping accent marks off of the OOV word and then copying it to the output (e.g. *caméra*, *éléments*, *molécules*). In the Science and EMEA domains, we found that our baseline BLEU scores improved from 21.91 to 22.20 and 23.67 to 24.45, respectively, when we changed the default handling of OOVs to strip accents before copying into the output. Interestingly, performance on the Subtitles domain text did not change at all with this baseline modification. This is likely due to the fact that there are fewer technical OOVs, which are the terms typically captured by this accent-stripping pattern, in the subtitles domain.

Throughout our experiments, we found it critical not to eliminate the potential to get such 'freebie' OOV translations correct by proposing an alternate, incorrect translation. In all of the results presented below, including the baselines, we supplement phrase tables with new candidate translations but also include accent-stripped identity, or 'freebie', translations in the table for all OOV words. We experimented with classifying French words as freebies or needing a new translation, but oracle experiments showed very little improvement (about 0.2 BLEU improvement in the Science domain), so instead of classifying words, we simply include both types of translations in the phrase tables.

In addition to the strip-accents baseline, we compare results with three additional baselines. First, we ranked English words[11] by their Levenshtein edit distance away from each French OOV word. Second, we ranked English words by their document-pair co-occurrence score (described in Section 11.3) with each French OOV word. Finally, we used the CCA model described in Daumé III & Jagarlamudi (2011) to rank English words according to their distributional similarity with each French word. Because of time constraints, we were only able to learn using 25,000 Science-domain document pairs, rather than the full 50,000 and for all domains, in the CCA baseline comparison. However, it's not likely that learning over more data would overcome the low performance observed so far. For each baseline, we include one new phrase table feature with the relevant translation score on new translation pairs and an indicator feature on accent-stripped pairs.

Table 18 shows results appending the top-1 and top-5 English translations for each OOV word using each of the baseline methods and for each domain. Interestingly, none of the alternate baselines outperform the simplest baseline on the subtitles data. Using document pair co-occurrences is the strongest baseline for the Science and EMEA domains.

[11] In particular, for each domain and each OOV French word, we ranked the set of all English words that appeared at least five times in the set of 50,000 most NEW-domain like Wikipedia pages. Using a frequency threshold of five helped eliminate French words and improperly tokenized English words from the set of candidates.

	Science	EMEA	Subs
Strongest Baseline	22.43	25.03	**13.18**
FF on new	**24.64**	**26.88**	13.06
FF on new + existing	24.56	26.54	12.93

Table 20: Comparison of BLEU score results (1) using the learned joint distribution to compute a feature function for appended phrase pairs only and (2) using it in combination with phrase-internal word alignments to also compute the feature function on existing phrase pairs. Top-1 translations are appended to source words with frequency of ten or fewer.

	Science	EMEA	Subs
Strongest Baseline	22.43	25.03	13.18
Mar. Match OOV	23.83	26.65	13.03
Oracle OOV	26.38	29.99	15.06
Poss. gain realized	35%	33%	-8%
Mar. Match freq<11	24.64	26.88	13.06
Oracle freq<11	27.91	31.82	16.03
Poss. gain realized	40%	27%	-4%

Table 21: BLEU score comparison of supplementing a phrase table with (1) strongest baseline reported in Table 18 for each domain, (2) marginal matching learned translations, and (3) oracle translations, derived from the word aligned development and test sets. We compare supplementing translations both for OOV words and for all source words appearing ten or fewer times in the training data.

This confirms our intuition that taking advantage of document pair alignments is worthwhile. In all cases, using the top-1 English translation outperforms using the top-5.

Tables 18, 19, and 20 show BLEU score results using learned translations to supplement the simple baseline phrase table. The tables show that adding only the top-1 translation for French words that appear with low frequency in the OLD-domain training corpus and using a constant value for new feature function on existing phrase pairs outperforms other experimental conditions.

Table 21 compares end-to-end MT performance when we supplement a baseline phrase table with our learned translations and when we supplement a baseline phrase table with oracle translations for the same set of source words. We compare adding translations for only OOV source words and for source words which appear ten or fewer times in the training data. Using the marginal matching learned translations takes us 40% of the way from the baseline to the oracle upper bound in the science domain and 27% of the way in the EMEA domain.

11.9 Discussion

BLEU score performance gains are substantial for the science and EMEA domains, but we don't observe any translation performance gains on the subtitles text. We believe this difference relates to the difference between a corpus domain and a corpus register. As Lee (2002) explains, a text's *domain* is most related to its topic, while a text's *register* is related to its type and purpose. For example, religious, scientific, and dialogue texts may be classified as separate registers, while political and scientific expositions may have a single register but different domains. Our science and EMEA corpora are certainly different in domain from the OLD-domain parliamentary proceedings, and our success in boosting MT performance with our methods indicates that the Wikipedia comparable corpora that we mined match those domains well. In contrast, the subtitles data differs from the OLD-domain parliamentary proceedings in both domain and register. Although the Wikipedia data that we mined may be closer in domain to the subtitles data than the parliamentary proceedings[12], its register is certainly not film dialogues.

Although the use of marginal matching is, to the best of our knowledge, novel in machine translation, there are related threads of research that might inspire future work. The intuition that we should match marginal distributions is similar to work using no example labels but only label proportions to estimate labels, for example in Quadrianto et al. (2008). Unlike that work, our label set corresponds to entire vocabularies, and we have multiple observed label

[12]In fact, we believe that it is. Wikipedia pages that ranked very high in our subtitles-like list included, for example, the movie *The Other Side of Heaven* and actor *Frank Sutton*.

OOVs translated correctly and incorrectly	
Input	les résistances au **cisaillement** par **poinçonnement** ...
Ref	the punching **shear strengths**...
Baseline	the resistances in **cisaillement** by **poinconnement** ...
MM	the resistances in **shear reinforcement**...
OOV translated incorrectly	
Input	présentation d' un logiciel permettant de gérer les données **temporelles** .
Ref	presentation of software which makes it possible to manage **temporal** data .
Baseline	introduction of a software to manage **temporelles** data .
MM	introduction of a software to manage data **plugged** .
Low frequency French words	
Input	...limite est liée à la **décroissance** très rapide du **couplage** électron-phonon avec la température .
Ref	...limit is linked to the rapid **decrease** of the electron-phonon **coupling** with temperature .
Baseline	...limit is linked to the **decline** very rapid electron-phonon **linkage** with the temperature .
MM	...limit is linked to the **linear** very rapid electron-phonon **coupling** with the temperature .

Table 22: Example MT outputs for Science domain. The baseline is the strip-accents baseline shown in Table 18, and the MM output corresponds to the Top-1 MM line in the same table. In the first example, the previously OOV word *cisaillement* is translated correctly by an MM-supplemented translation. The OOV *poinçonnement* is translated as *reinforcement* instead of *strengths*, which is incorrect with respect to this reference but arguably not a terrible translation. In the second example, *temporelles* is not translated correctly in the MM output. In the third example, the MM-hypothesized correct translation of low frequency word *couplage*, *coupling*, is chosen instead of the incorrect translation *linkage*. Also in the third example, the low frequency word *décroissance* is translated as the MM-hypothesized incorrect translation *linear*. In the case of *décroissance*, the baseline's translation, *decline*, is much better than the MM translation *linear*.

proportions. Also, while the marginal matching objective seems quite effective in practice, it is difficult to optimize. A number of recently developed approximate inference methods use a decomposition that bears a strong resemblance to this objective function. Considering the marginal distributions from each document pair to be a separate subproblem, we could approach the global objective of satisfying all subproblems as an instance of dual decomposition (Sontag et al., 2010) or ADMM (Gabay & Mercier, 1976; Glowinski & Marrocco, 1975). On the other hand, the incremental update of parameters also bears some resemblance to the margin infused relaxed algorithm (MIRA) (Crammer et al., 2006), where the divergence penalty is calculated between the current proposal and the last iteration's resulting value. Exploring variations of these optimization techniques may lead to faster convergence or better objective function values.

Our focus in this work has been on adapting an SMT model to translate text in some NEW-domain, and our methods may also be applicable to low-resource MT. In that setting, we can assume access to a standard dictionary or some small amount of seed parallel text (e.g. using crowdsourcing (Post et al., 2012)) from which we can estimate the old joint distribution, $p^{(\mathsf{old})}(s,t)$. The rest of the pipeline for learning new translations and supplementing an SMT model would look the same.

In the future, we plan to expand our model to learn multi-word translations. The main challenge will be, of course, the huge increase in the source and target vocabulary sizes. We could limit the phrase set using frequency, pointwise mutual information, or the joint distribution learned for lower order ngrams. For example, we may want to iterate over our comparable document pairs once to learn unigram translations and then again to learn bigram translations, using information from the first learning epoch.

11.10 Conclusions

We proposed a model for learning a joint distribution of source-target word pairs based on the idea that the distribution's marginals should match those observed in NEW-domain comparable corpora. Supplementing a baseline phrase-based SMT model with learned translations results in BLEU score gains of about two points in the medical and science domains.

12 Spotting New Senses

Previous analysis indicates that a significant percentage of translation errors in a new domain are "sense errors." (See Section 3.) That is, the word to be translated was observed in old domain training data (i.e. not OOV), but its correct translation was never learned. Given an old domain and a new domain, it would be useful to be able to identify words that gain at least one new translation in the new domain. New domain translations for these words could potentially be mined from comparable new domain data (Section 10) or retrieved directly from MTurkers through in-context translation tasks. Such words could also be a target for approaches involving active learning. The high frequency of sense errors in new domain translations suggests that specifcally targeting these words could greatly improve translation quality. Furthermore, the ability to identify words that gain new translations could help in assessing the difficulty of translating in any particular new domain.

To this end, we introduce the new binary classification task of "new sense spotting." For any word in the intersection of the old domain and new domain vocabularies, we would like to assign either a positive or negative label. A positive label for word w means that there exists at least one new domain translation for w that is never the translation of w in the old domain; a negative label means there are no new translations for w in the new domain.

12.1 Topic Model Feature

The intuition behind the topic model feature is that if a word gains a new translation in the new domain, its distribution over topics should change when moving into the new domain as well. For example, suppose that in our old domain, the word "run" is only ever translated as "courir," as in to run a marathon, and in our new domain, "run" may be translated either as "courir" or "execute," as in to run a computer program. Because of the new translation that "run" has gained in the new domain, the topic that places higher probability on related words like "computer," "program," and "executable," should also place a higher probability on the word "run." In the old domain, however, we would not expect a similar topic (if it exists) to give a higher probability to the word "run." Thus we compute the following score:

score(w) = $\sum_{k \in topics_{new}} P_{new}(k|w) \times \sum_{k' \in topics_{old}} (P_{old}(k'|w) \times cossim(k, k'))$

For any given word, w, this score will be higher if, for each new domain topic, k, that places high probability on w, there is an old domain topic, k', that has a high cosine similarity to k and also places a high probability on w. Conversely, if no such topic exists, the score will be lower, perhaps indicating the word has gained a new sense in the new domain.

12.2 Fill-in-the-Blank Feature

Another feature we use to predict whether a word, w, gains a new sense or not is the Jenson-Shannon divergence between two probability distributions over candidate translations of w, as learned from old domain data and new domain data, respectively. First we build a new-domain classifier that predicts a target-side word given sentence context, represented as a bag of words (and the spelling features thereof). The data for this classifier is target-side new domain monoligual (or comparable) text. The computation proceeds as follows:

- For each word, w, in the source language
  - Generate a set of translation candidates C(w) from all available parallel data (mostly old domain, some new domain).
  - For each available source-side sentence, s, (e.g. from comparable or monolingual data) that contains w
    - Remove w from s. Use old domain language model to translate each unigram remaining in the sentence to target side, generating a bag of words feature (plus spelling features).
    - Use the classifier described above to get a distribution over the set C(w).
    - Compute JS divergence over the set, C(w), between the distributions predicted by the old domain language model and the new classifier.
- Return average JS divergence over all sentences.

12.3 N-Gram Feature

Another indicator that a word has perhaps gained a new translation in the new domain is to look at that word's closely neighboring words. It is expected that words acquiring new senses will tend to neighbor different sets of words (e.g. different arguments, prepositions, parts of speech, etc.). Thus, the n-gram feature is merely the ratio of the number of new domain n-grams (up through trigrams) containing word w to the total number of new domain n-grams cotaining w. This is done at the type level. Additionally, n-grams containing OOV words are not counted, as they may simply be an instance of applying the same sense of a word to a new argument (e.g. a proper noun not seen in the old domain).

$$\text{n-gram-score(w)} = \frac{|\{new-domain-ngrams-containing-w-not-found-in-old-domain\}|}{|\{all-new-domain-ngrams-containing-w\}|}$$

12.4 Results

We tested the above features on a new sense spotting classification task. With no features added, the new sense spotter correctly classifies half of the examples (a score of 0.5). Below are results for topic model-based and n-gram-based features.

Table 23: Feature Results

Domain	Topic Model	N-Gram	Both
EMEA	0.586	0.603	0.657
Science	0.564	0.523	0.631
Subs	0.473	0.575	0.574

13 Latent Topics as Domain Indicators

13.1 Introduction

In this section, we consider methods that leverage document-level information in the MT task. As a motivating example, consider translating the sentence "He couldn't find a **match**." This sentence provides little guidance on how to translate the word 'match', which could be either a small instrument used to start a fire, or a correspondence between two types of objects. Whether we include word-based, phrasal, or even very long-distance features including syntax or argument structure, the system does not have sufficient information to pick the proper translation. However, if we know that the topic of the document relates to finding medical documents (e.g. transplant donors) rather than starting fires, the system may be able to predict the appropriate translation.

Indeed, previous work has shown that both explicit document-level information such as document provenance (Chiang et al., 2011) and implicit domain indicators such as topic models (Eidelman et al., 2012) can be helpful for the MT task. We investigated applying this type of information to the domain adaptation setting. In particular, we are interested in whether adding document-level information to the MT model will be useful in the two different data cases. First, we consider the case in which we lack parallel data in the new domain but we do have monolingual source data in the new domain. Next, we consider the case in which there is some parallel data in the new domain, and we wish to take full advantage of it in informing our topic modeling for the MT task.

13.2 Latent Topic Models

Following prior work (Eidelman et al., 2012), we start with a LDA topic model, in which each document d_i is represented by a mixture of topics z_n. Associated with each topic z_n is a probability distribution generating words $p(w_i|z_n, \beta)$. Given a set of documents, this model learns one topic distribution for each document and a global set of word distribution for each topic to optimize the likelihood of the full dataset.

13.3 Lexical Weighting Models

To address the first setting, in which for the new domain we have monolingual data only, we built generative latent topic models over the source data. The resulting topic distributions were used to create lexical weighting models that were used in the translation model directly. They could inform as features for the PSD classifier (see Ch. 5).

Using these topic models, we explored two types of lexical weighting models: conditioning on either the document-level distribution or the document- and token-level posterior distribution. Since Eidelman et al. (2012) found that more peaked document-level topic distributions were most helpful for MT, we created these additional lexical weighting models that use the per-word posterior distribution over topics with the idea that this might lead to a sharper and more helpful model. We estimated the document-topic-conditioning lexical weighting models according to the following criteria: For aligned word pair (e, f), compute the expected count $e_{z_n}(e, f)$ under topic z_n:

$$e_{z_n}(e, f) = \sum_{d_i \in T} p(z_n|d_i) \sum_{x_j \in d_i} c_j(e, f)$$

Then compute the lexical probability conditioned on the topic distribution:

$$p_{z_n}(e, |f) = \frac{e_{z_n}(e, f)}{\sum_e e_{z_n}(e, f)}$$

For the token-topic-conditioning models, we add the additional conditioning context of the source word:

$$e_{z_n}(e, f) = \sum_{d_i \in T} \sum_{x_j \in d_i} \sum_{f_k \sim e \in x_j} p(z_n|d_i, f_k),$$

where

$$p(z_n|d_i, f_k) \propto p(f_k|z_n) \cdot p(z_n|d_i).$$

These lexical weighting models could then be added as a feature in a log-linear translation model. We compute the lexical weight over all words in a phrase and use it as a feature in phrase-based translation:

$$f_{z_n}(\bar{e}|\bar{f}) = -\log\{p_{z_n}(\bar{e}, |\bar{f})p(z_n|d)\}$$

$$\sum_p \lambda_p h_p(\bar{e}, \bar{f}) + \sum_{z_n} \lambda_{z_n} f_{z_n}(\bar{e}|\bar{f}).$$

13.4 Discriminative Latent Variable Topics

In this section, we consider the case in which there is parallel data available in the new domain. The traditional topic models seen in section 13.3 are monolingual: they find a mixture of unigram distributions that optimizes the likelihood of some monolingual document set. However, the topics only look at one side of the parallel data. Intuitively it would make sense for the MT task to learn topics that leverage both languages. Several approaches have been suggested for so-called polylingual topic distributions (Mimno et al., 2009; Platt et al., 2010; Jagarlamudi & III, 2010). These approaches generally try to model the joint likelihood of both documents.

For the MT task, though, we might prefer a model of the *conditional* likelihood of the target language given the source, as this is the goal we hope to achieve. Furthermore, there are several limitations of the generative topic models that we would like to address.

A first limitation is that each word gets an equal voice in selecting the topic distribution of the document. In a conventional LDA topic model, the probability of a document is

$$P(\theta|\alpha)\prod_i P(z_i|\theta)P(w_i|z_i).$$

The posterior distribution over topics looks like a naïve Bayes model given the words: each word gets equal weight in selecting topics. This is unfortunate. Many common words (such as "*the*" or "*is*") have no strong preference amongst topics. They translate in the same way regardless of topic. Other words may be strong indicators of a particular topic (such as "*the*" versus "*hexachordal*").

A second limitation is that each topic learns a totally independent distribution. In practice, some words translate in different ways depending on the topic (such as "*bank*"); others are more consistent across varying contexts (such as "*the*"). We would like a model that addresses this with sharing.

Our idea here is to replace the generative model with a discriminative one that optimizes likelihood directly. First, we predict the probability of each topic using a log-linear model with features from the whole source document. This allows some words to vote strongly for particular topics, and others to quietly vacillate without influencing the distribution substantially.

Second, we replace the translation distribution with another log-linear distribution. We assume that there are 2^B topics for some value B, and that they live in a simple hierarchy consisting of a binary tree. Say we have $B = 2$, so there are four topics. Then in addition to the leaf topics $0, 1, 2$, and 3, we add three "super-topics": $\{0, 1\}, \{2, 3\}$, and $\{0, 1, 2, 3\}$. When extracting features for, say, topic 2, we also emit features for the super-topics $\{2, 3\}$ and $\{0, 1, 2, 3\}$. This allows the parameter estimation procedure to set parameters at the appropriate level of the hierarchy. Words that do not depend on the topic assignment may have most of their parameters set of the root of the topic tree. Other words that are influenced by context may learn parameters at lower levels.

We devote the remainder of this section to the formal description of the model and estimation of its parameters.

13.4.1 Notation

- $\mathbf{\Sigma}, \mathbf{T}$: source and target language vocab
- S: Source language document
- T: Target language document
- s, t: source and target language words

- K: number of topics
- $Z = \{z_1, \ldots, z_k\}$: topics
- $F : 2^{\Sigma} \times Z \rightarrow R^m$: topic feature function
- $G : S \times Z \times T \rightarrow R^n$: translation feature function
- $\theta \in R^m, \phi \in R^n$: parameter vectors for topics and translations, respectively

13.4.2 Model

We aim to model the conditional likelihood of a target document given a source document, using a mixture of latent topics:

$$P(T|S) = \sum_{z \in Z} \left(P(z|S) \prod_{(s,t) \in (S,T)} P(t|s,z) \right)$$

The topic distribution is predicted based on features of the whole source document:

$$P(z|S) \propto \exp(\theta \cdot F(S,z))$$

Each translation is predicted based only on the source word and a given topic likelihood:

$$P(t|s,z) \propto \exp(\phi \cdot G(s,z,t))$$

So the likelihood, expanded out, is as follows:

$$P(T|S,\theta,\phi) = \sum_{z \in Z} \left(\frac{\exp(\theta \cdot F(S,z))}{Z_\theta} \prod_{(s,t) \in (S,T)} \frac{\exp(\phi \cdot G(s,z,t))}{Z_\phi} \right)$$

Here is the log likelihood of a single target document Note that the sum over mixture components prevents the log from further advances, unlike standard logistic regression models.

$$\log P(T|S,\theta,\phi) = \log \sum_{z \in Z} \left(\frac{\exp(\theta \cdot F(S,z))}{Z_\theta} \prod_{(s,t) \in (S,T)} \frac{\exp(\phi \cdot G(s,z,t))}{Z_\phi} \right)$$

13.4.3 Partial Derivatives for Components of θ

First let us focus on computing the gradient of the topic distribution.

$$\frac{\partial}{\partial \theta_i} \left[\log P(T|S,\theta,\phi) \right] = \frac{\partial}{\partial \theta_i} \left[\log \sum_{z \in Z} \left(\frac{\exp(\theta \cdot F(S,z))}{Z_\theta} \prod_{(s,t) \in (S,T)} \frac{\exp(\phi \cdot G(s,z,t))}{Z_\phi} \right) \right]$$

We have that $\frac{d}{dx} \left[\log(f(x)) \right] = \frac{1}{f(x)} \frac{df(x)}{dx}$

$$= \frac{1}{P(T|S,\theta,\phi)} \frac{\partial}{\partial \theta_i} \left[\sum_{z \in Z} \left(\frac{\exp(\theta \cdot F(S,z))}{Z_\theta} \prod_{(s,t) \in (S,T)} \frac{\exp(\phi \cdot G(s,z,t))}{Z_\phi} \right) \right]$$

Push inside the sum

$$= \frac{1}{P(T|S,\theta,\phi)} \sum_{z \in Z} \left(\frac{\partial}{\partial \theta_i} \left[\frac{\exp(\theta \cdot F(S,z))}{Z_\theta} \prod_{(s,t) \in (S,T)} \frac{\exp(\phi \cdot G(s,z,t))}{Z_\phi} \right] \right)$$

Push across the prediction portion, which is constant with respect to θ:

$$= \frac{1}{P(T|S,\theta,\phi)} \sum_{z \in Z} \left(\frac{\partial}{\partial \theta_i} \left[\frac{\exp(\theta \cdot F(S,z))}{Z_\theta} \right] \prod_{(s,t) \in (S,T)} P(t|s,z,\phi) \right)$$

Quotient rule: $\frac{d}{dx}\left[\frac{f(x)}{g(x)}\right] = \frac{\frac{df(x)}{dx}g(x) - f(x)\frac{dg(x)}{dx}}{(g(x))^2}$. In this case, $f(x)$ is a density of exponential form so $f(x) = exp(h(x))$, and $g(x)$ is a partition function. Thus, the first term is the probability times the derivative of the density before exponentiating: $\frac{\frac{df(x)}{dx}gx}{(g(x))^2} = \frac{\frac{df(x)}{dx}}{g(x)} = \frac{\frac{d\exp h(x)}{dx}}{g(x)} = \frac{\exp h(x)\frac{dh(x)}{dx}}{g(x)} = \frac{f(x)\frac{dh(x)}{dx}}{g(x)} = p(x)\frac{dh(x)}{dx}$. Regarding the second term, that turns into a probability times an expectation

$$= \frac{1}{P(T|S,\theta,\phi)} \sum_{z \in Z} \left(\left(P(z|S,\theta) \cdot F_i(S,z) - P(z|S,\theta) \sum_{z'} P(z'|S,\theta) F_i(S,z') \right) \prod_{(s,t) \in (S,T)} P(t|s,z,\phi) \right)$$

$$= \frac{1}{P(T|S,\theta,\phi)} \sum_{z \in Z} \left(P(z|S,\theta) \left(F_i(S,z) - \mathbb{E}_{z'|S,\theta}\left[F_i(S,z')\right] \right) \prod_{(s,t) \in (S,T)} P(t|s,z,\phi) \right)$$

13.4.4 Neat Trick

Taking the derivative of a complex product can lead to many terms. Luckily there is a compact way to represent this product using some calculus:

$$\begin{aligned}
&\frac{\partial}{\partial \theta}\left[\prod_i f_i(\theta)\right] \\
&= \sum_i \prod_{j \neq i} f_j \frac{\partial}{\partial \theta}\left[f_i\right] \\
&= \sum_i \frac{1}{f_i} \prod_j f_j \frac{\partial}{\partial \theta}\left[f_i\right] \\
&= \sum_i \left(\prod_j f_j \right) \frac{1}{f_i} \frac{\partial}{\partial \theta}\left[f_i\right] \\
&= \left(\prod_j f_j \right) \sum_i \frac{1}{f_i} \frac{\partial}{\partial \theta}\left[f_i\right]
\end{aligned}$$

But we know that $\frac{\partial}{\partial \theta}\left[\log f\right] = \frac{1}{f}\frac{\partial}{\partial \theta}\left[f\right]$, so we get:

$$\frac{\partial}{\partial \theta}\left[\prod_i f_i(\theta)\right] = \left(\prod_i f_i \right) \sum_i \frac{\partial}{\partial \theta}\left[\log f_i\right]$$

This is a much nicer expression to work with, especially in our log-linear models.

13.4.5 Partial Derivatives for Components of ϕ

$$\frac{\partial}{\partial \phi_i}\left[\log P(T|S,\theta,\phi)\right] = \frac{\partial}{\partial \phi_i}\left[\log \sum_{z \in Z} \left(\frac{\exp(\theta \cdot F(S,z))}{Z_\theta} \prod_{(s,t) \in (S,T)} \frac{\exp(\phi \cdot G(s,z,t))}{Z_\phi} \right) \right]$$

We have that $\frac{d}{dx}\left[\log(f(x))\right] = \frac{1}{f(x)}\frac{df(x)}{dx}$

$$= \frac{1}{P(T|S,\theta,\phi)} \frac{\partial}{\partial \phi_i} \left[\sum_{z \in Z} \left(\frac{\exp(\theta \cdot F(S,z))}{Z_\theta} \prod_{(s,t) \in (S,T)} \frac{\exp(\phi \cdot G(s,z,t))}{Z_\phi} \right) \right]$$

Push inside the sum

$$= \frac{1}{P(T|S,\theta,\phi)} \sum_{z \in Z} \left(\frac{\partial}{\partial \phi_i} \left[\frac{\exp(\theta \cdot F(S,z))}{Z_\theta} \prod_{(s,t) \in (S,T)} \frac{\exp(\phi \cdot G(s,z,t))}{Z_\phi} \right] \right)$$

Now the first term is constant

$$= \frac{1}{P(T|S,\theta,\phi)} \sum_{z \in Z} \left(P(z|S,\theta) \frac{\partial}{\partial \phi_i} \left[\prod_{(s,t) \in (S,T)} \frac{\exp(\phi \cdot G(s,z,t))}{Z_\phi} \right] \right)$$

Then apply our trick

$$= \frac{1}{P(T|S,\theta,\phi)} \sum_{z \in Z} \left(P(z|S,\theta) \left(\prod_{(s,t) \in (S,T)} P(t|s,z,\phi) \right) \sum_{(s,t) \in (S,T)} \frac{\partial}{\partial \phi_i} \left[\log \frac{\exp(\phi \cdot G(s,z,t))}{Z_\phi} \right] \right)$$

$$= \frac{1}{P(T|S,\theta,\phi)} \sum_{z \in Z} \left(P(z|S,\theta) \left(\prod_{(s,t) \in (S,T)} P(t|s,z,\phi) \right) \sum_{(s,t) \in (S,T)} \left(G_i(s,z,t) - \mathbb{E}_{t'|s,z} \left[G_i(s,z,t') \right] \right) \right)$$

13.4.6 Complete Gradient

Consider a posterior distribution over each topic for a given document pair, defined as this:

$$P(z|S,T,\theta,\phi) = \frac{P(T,z|S,\theta.\phi)}{P(T|S,\theta,\phi)}$$

Then the gradients are just differences between empirical counts and true counts modulated by expectation under this distribution. So we have:

$$\frac{\partial}{\partial \theta_i} \left[\log P(T|S,\theta,\phi) \right] = \mathbb{E}_{z \sim |S,T,\theta,\phi} \left[F_i(S,z) - \mathbb{E}_{z'|S,\theta} \left[F_i(S,z') \right] \right]$$

$$\frac{\partial}{\partial \phi_i} \left[\log P(T|S,\theta,\phi) \right] = \mathbb{E}_{z \sim |S,T,\theta,\phi} \left[\sum_{(s,t) \in (S,T)} \left(G_i(s,z,t) - \mathbb{E}_{t'|s,z,\phi} \left[G_i(s,z,t') \right] \right) \right]$$

Now for each document, we can first compute the log density of each topic under the current model, and normalize to get a distribution. Then we gather statistics and multiply by this posterior as necessary.

13.4.7 Optimization

Currently we're exploring batch optimization using RProp (Calandra, 2011). Note that we have to initialize with a random vector. A zero vector sits on a saddle point with respect to likelihood, and the model fails to make progress if we start there. We incorporate an L2 norm as a regularizer, but its weight must be small (no larger than 0.1) to allow sharp topic distributions.

13.4.8 Simple Example

Say we have a corpus containing the following two French-English sentence pairs, where the subscript indicates the word alignment:

(1a) le_1 régime_2 démocratique_3

(1b) the_1 democratic_3 regime_2

(2a) le_1 régime_2 pamplemousse_3

(2b) the_1 grapefruit_3 diet_2

If we learn two latent topics, the resulting document distribution looks like this:

	topic 0	topic 1
sentence 1	0.01	0.99
sentence 2	0.99	0.01

The translation distribution is:

source	target	super-topic01	topic0	topic1
le	the	1.00	1.00	1.00
régime	regime	0.45	0.99	0.01
régime	diet	0.55	0.01	0.99
démocratique	democratic	1.00	1.00	1.00
pamplemousse	grapefruit	1.00	1.00	1.00

Clearly *démocratique* and *pamplemousse* are able to significantly influence the topic distribution, even though their translation is not topic dependent. Also note that the translation of *régime* is successfully disambiguated by the topic indicator.

Of course, this simple example could also be clearly learned by a phrasal model. The broader point is that we can capture more global relationships with this model.

13.5 Experimental Setup

We used the canadian hansards as the old-domain and EMEA as the new domain. We trained the topic models on the top 5,000 most frequent words in each corpus. We used vw to learn topic models for the old and new domain data. We built topic models using 5, 10, 15, and 20 topics, with the α parameter set to 0.1 and 0.01. For the alignment data, we used the first 250,000 sentences of each corpus.

We considered 3 different settings for the alignment data and the topic model data: old-alignment/old-topics, new-alignments/new-topics, and old-alignments/new-topics.

13.6 Evaluation

For intrinsic evaluation of the lexical weighting models, we compared the average log likelihood of a held-out set of new-domain data, consisting of the last 5,000 sentences of the EMEA MT training data. Table 24 summarizes the results. We see that the conditioning on latent topics helps the log likelihood in all settings, though conditioning on

	no-topic	doc-topic	word-topic
old-alignment, old topic	-1.78	-0.47	-0.48
new-domain, new-topic	-1.12	-0.26	-0.26
old-domain, new-topic	-1.78	-0.27	-0.27

Table 24: Average per-word log likelihood of EMEA data

token as well as document topic distribution does not make a difference. We can also note that using a new-domain topic model with old-domain alignments allows us to get as good a log-likelihood on the data as using new-alignments with new-topic. This is encouraging for using these models in MT in the setting in which we have have old-domain parallel data but only monolingual new-domain data.

To use the lexical weighting models in the MT system, we added the lexical weights into the pre-trained phrase table. Results are forthcoming.

The discriminative topic models are still training and results are forthcoming.

13.7 Future Work

We are currently training phrase-based MT systems using the generative and discriminative topic model features; following that, we plan to also use these features in a hierarchical phrase-based systems. We would also like to introduce the topics model features into the PSD classifier, since this may allow interaction between longer distance features than those considered by the phrase-based decoder.

14 Mining Token Level Translations Using Dimensionality Reduction

SMT systems rely on word/phrase level translation tables to translate sentences from one language into another. The translation table lists possible translations (e.g. Table 25) of a word irrespective of the context in which the word has occurred. For concreteness, we call them type level translations since they are independent of the word's context. In this section, we address the problem of re-scoring type level translations based on the word's context. For e.g., given the French sentence "Il a rédigé un rapport" (whose English translation is "He wrote a report") we want to score the English translation 'report' higher, where as given the sentence "Quel est le rapport" (What is the relationship) we want to assign high score to the translation 'relationship'. We refer to this problem as adapting type level translations to the token level.

French	English	$p(e\|f)$
rapport	report	0.3
rapport	document	0.3
rapport	relationship	0.1
rapport	reporting	0.05

Table 25: Type level translations of the French word rapport.

Mining token level translations can interact with SMT in the following three different ways: 1) to mine translations in the new sense, i.e. once a French word has been identified as it is used in a new sense (Sec. 12) we can use our approach to mine the translations in the new sense. 2) it can be fed as an additional feature for phrase sense disambiguation (PSD) classifier (Sec. 5) and 3) this can be used to gather new training instance for the PSD classifier. PSD classifier is trained for *only* the source and target language pairs that are observed in the translation table, but we can address this limitation by including additional source and target language translation pairs that are mined by our approach.

We want our approach to handle out-of-vocabulary (OOV) words and also identify the translations in the new sense. For example, in the scientific domain, 'rapport' translates 'ratio' which is not observed in the old domain data. Since we want our approach to handle these cases as well, we extend on the existing idea of representing words in an interlingual representation which showed promise in mining translations for the OOV words (Haghighi et al., 2008; Daumé III & Jagarlamudi, 2011). Our approach involves two main steps: 1) learning type vectors, i.e. representing source and target language words in a k-dimensional interlingual representation. 2) learning to adapt the type vectors to the token level. We will describe both these steps in detail in the following two sections, but before that we fix the terminology that is used in this section.

14.1 Notation

In general, a bold lower case letter ($\boldsymbol{x}$) represents a column vector, an upper case letter (X) represents a matrix and a greek letter represents a function. Let m_f and m_e represent the vocabulary size in French and English languages respectively. Let $\phi(.)$ and $\psi(.)$ represent feature functions which take a French and English word respectively and output a feature vector. Let $\boldsymbol{x}_i \in \mathbb{R}^{d_1}$ and $\boldsymbol{y}_j \in \mathbb{R}^{d_2}$ represent the feature vectors of the French and English words f_i and e_j respectively, i.e. $\boldsymbol{x}_i = \phi(f_i)$ and $\boldsymbol{y}_j = \psi(e_j)$. Moreover, let $\zeta(.)$ and $\eta(.)$ be functions that take a French and English word respectively and return their lower k-dimensional vectors, i.e. $\zeta(f_i) \in \mathbb{R}^k$ and $\eta(e_i) \in \mathbb{R}^k$. We differentiate the lower dimensional type and token level embeddings with subscripts p and k respectively.

14.2 Learning Type Vectors

We use canonical correlation analysis (CCA) (Hotelling, 1936) to learn the interlingual representation (Daumé III & Jagarlamudi (2011)). It uses n word translation pairs as training data to learn the interlingual representation. As mentioned above, each word is represented as a feature vector. Let $X(d_1 \times n)$ and $Y(d_2 \times n)$ represent the data matrices with word feature vectors as the columns. Notice that the input feature vectors for French and English words are of different lengths, i.e. d_1 and d_2 respectively, so their feature spaces are completely different. Finding an interlingual representation is an attempt to map both source and target language words into a common sub-space which facilitates us to learn the token specific vectors. In this section, we assume that the input feature functions $\big(\phi(.)$ and $\psi(.)\big)$ are

known and describe the use of CCA to learn the interlingual representation. Subsequently, we describe the feature functions.

Given a multi-view data, Canonical Correlation Analysis (Hotelling, 1936) is a technique to find the projection directions in each view so that the objects when projected along these directions are maximally aligned. Let X ($d_1 \times n$) and Y ($d_2 \times n$) be the representation of n-word pairs in both the languages respectively, then CCA finds the projection directions $\boldsymbol{a}$ and $\boldsymbol{b}$ such that.

$$\arg\max_{\boldsymbol{a},\boldsymbol{b}} \frac{\boldsymbol{a}^T X Y^T \boldsymbol{b}}{\sqrt{\boldsymbol{a}^T X X^T \boldsymbol{a}}\sqrt{\boldsymbol{b}^T Y Y^T \boldsymbol{b}}}$$

$$\arg\min_{\boldsymbol{a},\boldsymbol{b}} ||X^T \boldsymbol{a} - Y^T \boldsymbol{b}||^2 \qquad \text{s.t. } \boldsymbol{a}^T X X^T \boldsymbol{a} = 1 \ \& \ \boldsymbol{b}^T Y Y^T \boldsymbol{b} = 1$$

The projection directions are obtained by solving the eigen system:

$$\begin{bmatrix} 0 & C_{xy} \\ C_{yx} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} = \lambda \begin{bmatrix} C_{xx} & 0 \\ 0 & C_{yy} \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix}$$

where C_{xx}, C_{yy} are the covariance matrices for X and Y and C_{xy} is the cross-covariance. The projection directions of regularized CCA solves are the eigen vectors of

$$\begin{bmatrix} 0 & C_{xy} \\ C_{yx} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} = \lambda \begin{bmatrix} C_{xx} + \lambda I & 0 \\ 0 & C_{yy} + \lambda I \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix}$$

In general, using all the eigen vectors is sub optimal and thus retaining top eigen vectors leads to an generalizability. So, let $A \in \mathbb{R}^{(d_1 \times k)}$ and $B \in \mathbb{R}^{(d_2 \times k)}$ be the projection directions with the top k eigenvectors $\boldsymbol{a}$ and $\boldsymbol{b}$ as columns respectively. Then the low-dimensional type level embedding of a French word f_i is given by $\zeta_p(f_i) = A^T \phi(f_i)$, where $\phi(f_i)$ returns the feature vector of the French word. Similarly, the low-dimensional type level embedding of an English word e_j is given by $\eta_p(e_j) = B^T \psi(e_j)$.

14.3 Features

In the previous section, we skipped the description of the feature functions and the selection of n-word pairs used for training. We will describe these two aspects in this section.

From the target domain corpus we extract the most frequent words for both the languages. Of these, words that have translation in the bilingual dictionary (learnt from Hansards) are used as training data. First, we extract feature vectors for all the words. We use context and orthographic features. Second, using the translation probabilities of seen words, we identify wordpairs whose feature vectors are used to learn the CCA projection directions. Finally, the type vectors are obtained by projecting all the words into the sub-space identified by CCA as described towards the end of the previous 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. 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 pre-processng for all words.

Since a word can have multiple translations, and that CCA needs 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 & 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 to form the projection directions A and B. In other relevant experiments, we have found that this setting of CCA outperforms the baseline approach.

14.4 From Type to Token Level Embeddings

In this section, we describe our approach to adapt type level word embeddings to the token level. For clarity, we use the French word 'rapport' as the running example. Table 26 shows two different contexts in which the word occurred and also shows the target language translation in both the cases. We use such word aligned parallel data as training data for the token level adaptation. Notice that the words in the context give an indication of the target translation, e.g. the cue word 'rédigé' in the first sentence is a good indicator that the translation could be 'report'. We refer to the word that we are adapting as the *focus* word and the words in its context as *cue* words. We limit the cue words to be a window of words around the focus word.

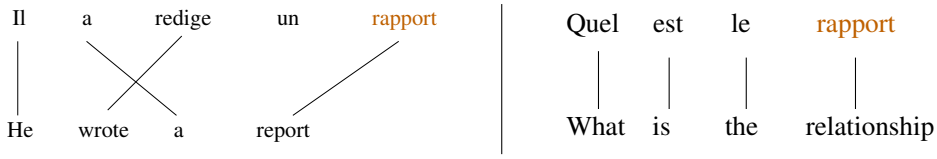


Table 26: Two different contexts/tokens of the word 'rapport'. Notice that the word translates into different English words depending on the context.

As described above, we first use CCA to get the type vectors for all the words. Then the token vector of the focus word is assumed to be a weighted linear combination of the cue word type vectors.[13] Formally, given a French word f_i its token vector $\zeta_t(f_i)$ is given as:

$$\zeta_t(f_i) = w_0\zeta_p(f_i) + \sum_{f_j \in \mathcal{N}(f_i)} w_{f_j}\zeta_p(f_j) \tag{3}$$

where $\zeta_p(.)$ is a function that returns the type vector of a French word, $\mathcal{N}(f_i)$ returns the cue words that are in the neighbourhood of the focus word f_i, w_{f_j} is the contribution of the cue word f_j towards adapting the focus word and w_0 is a special weight which indicates the contribution of the focus word towards itself. Intuitively, we would expect the type and token vectors of a word to be closer so we would expect w_0 to be higher than the other weights.

We use word aligned parallel data to learn the weight vector. For each French token, we also know its English translation. So, we want to find the weight vector such that the token vector of the French word is closer to the type vector of the English word.[14] This can be expressed using the following objective function:

$$\underset{w_0,\boldsymbol{w}}{\operatorname{argmin}} \sum_i \left|\left|\zeta_t(f_i) - \eta_p(e_i)\right|\right|^2 \tag{4}$$

$$\underset{w_0,\boldsymbol{w}}{\operatorname{argmin}} \sum_i \left|\left|\left(w_0\zeta_p(f_i) + \sum_{f_j \in \mathcal{N}(f_i)} w_{f_j}\zeta_p(f_j)\right) - \eta_p(e_i)\right|\right|^2 \tag{5}$$

Intuitively, we are using the cue words to capture the contextual information. And the target language translations provide additional information about the sense in which the focus word is used in each of the contexts. The idea is that all the contexts in which the focus word translates to the same word should modify the type vector in the same way.

If we consider a window of length two words around the focus word, then the token vector of 'rapport' in the above two running examples is given as:

$$w_0\zeta_p(\text{rapport}) + w_{\text{rédigé}}\zeta_p(\text{rédigé}) + w_{\text{un}}\zeta_p(\text{un}) \tag{6}$$

$$w_0\zeta_p(\text{rapport}) + w_{\text{est}}\zeta_p(\text{est}) + w_{\text{le}}\zeta_p(\text{le}) \tag{7}$$

And the weights are learned such that the resulting token vectors are close to the type vectors $\eta_p(\text{report})$ and $\eta_p(\text{relationship})$ respectively. Thus the cue words help us differentiate the different senses of the focus word.

14.4.1 Optimization

Since the adaptation function (Eq. 3) is linear in terms of the weight vector, it can rewritten as a matrix-vector product which will become useful when we try to learn the weights. Let Z be a $(k \times m_f)$ matrix which stores the type vectors

[13] Why weighted linear combination, argument from the compositionality literature.

[14] Token and Type rather than Token vectors in both the languages.

of all the French words. The i^{th} column of Z stores the type vector of the French word f_i. Let I_i be a $m_f \times (m_f + 1)$ indicator matrix. The first column of this matrix indicates the word type of the focus word and the rest of the columns indicate the context words around the focus word.

$$I_i(1,j) = \begin{cases} 1, & \text{if focus word is } f_j \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

$$I_i(j+1,j) = \text{frequency of } f_j \text{ in the window around the focus word} \quad (9)$$

The rest of the elements of this matrix are set to zero, hence this is a very sparse matrix. Let $\tilde{\boldsymbol{w}} = [w_0 \;\; \boldsymbol{w}^T]^T$ then the adaptation function in Eq. 3 can be rewritten as $\zeta_t(f_i) = Z\, I_i\, \tilde{\boldsymbol{w}}$ and the objective function in Eq. 5 can be rewritten as follows:

$$\underset{\tilde{\boldsymbol{w}}}{\operatorname{argmin}} \sum_i \mid ZI_i\tilde{\boldsymbol{w}} - \eta_p(e_i) \mid^2 \quad (10)$$

Differentiating and setting the derivative with respect to $\tilde{\boldsymbol{w}}$ will result in the following linear system of equations which can be solved very efficiently.[15]

$$\Big(\sum_i I_i^T Z^T Z I_i\Big)\tilde{\boldsymbol{w}} = \sum_i I_i^T Z^T \eta_p(e_i) \quad (11)$$

14.4.2 Co-Regularization

In the previous adaptation model, the weight of a cue word depends only on the cue word and not on the focus word. A cue word such as 'rédigé' (wrote) can be a good indicator for the focus word 'rapport', but it may not be such a good indicator for a different focus word such as 'premier' (whose possible translations are prime minister, first day). Here we propose an extension of the previous model where the weight depends on the cue word and the focus word. In other words, there is a weight vector specific to every focus word.

$$\zeta_t(f_i) = ZI_i\tilde{\boldsymbol{w}}_{f_i}$$

But at the same time, it introduces many parameters into the model and may lead to data sparsity problem. In order to overcome the sparsity issue we tie the all the weight vectors by a common weight vector ($\tilde{\boldsymbol{w}}$), i.e. we assume $\tilde{\boldsymbol{w}}_{f_i} \leftarrow \tilde{\boldsymbol{w}} + \tilde{\boldsymbol{r}}_{f_i}$, and try to minimize the residual vector $\tilde{\boldsymbol{r}}_{(.)}$ as much as possible. The objective function under this model is expressed as follows:

$$\underset{\tilde{\boldsymbol{w}},\tilde{\boldsymbol{r}}_{(.)}}{\operatorname{argmin}} \sum_i \mid ZI_i(\tilde{\boldsymbol{w}} + \tilde{\boldsymbol{r}}_{f_i}) - \eta_p(e_i) \mid^2 + \lambda \sum_{f_j} \mid \tilde{\boldsymbol{r}}_{f_j} \mid^2 \quad (12)$$

14.4.3 Discriminative Adaptation

Both the above models only use the target translation of a French word and ignore other candidate possible translations. For example, in the first running example the previous models only use the fact that the word 'rapport' translates to 'report' but they ignore the fact that there are other candidate translations ('document', 'relationship', etc.) and that 'report' is a better choice than the remaining candidate translations. In this model, we explicitly use this information. The aim is to learn a weight vector such that the token vector is closer to the correct translation but is farther from the other candidate translations.

Before formulating the new objective function, we slightly rewrite the objective function shown in Eq. 10 as follows:

$$\underset{\tilde{\boldsymbol{w}}}{\operatorname{argmin}} \sum_i \mid ZI_i\tilde{\boldsymbol{w}} - \eta_p(e_i) \mid^2 \quad (13)$$

$$\underset{\tilde{\boldsymbol{w}}}{\operatorname{argmax}} \sum_i 2\,\tilde{\boldsymbol{w}}^T I_i^T Z^T \eta_p(e_i) - \sum_i \tilde{\boldsymbol{w}}^T I_i^T Z^T Z I_i \tilde{\boldsymbol{w}} \quad (14)$$

[15] The left hand side of this equation can be computed efficiently as the element wise product of $Z^T Z$ and the co-variance of the indicator matrices. It is efficient in terms of both space and time compared to its naive implementation.

We add the discriminative term to the above function such that the token vector moves away from the other candidate translations. The resulting new objective function is given by:

$$\underset{\tilde{\boldsymbol{w}}}{\operatorname{argmax}} \sum_i 2\, \tilde{\boldsymbol{w}}^T I_i^T Z^T \eta_p(e_i) - \sum_i \tilde{\boldsymbol{w}}^T I_i^T Z^T Z I_i \tilde{\boldsymbol{w}} + \mu \sum_i \sum_{e_j \in trans(f_i)\ \&\ e_i != e_j} \tilde{\boldsymbol{w}}^T I_i^T Z \big(\eta_p(e_i) - \eta_p(e_j)\big) \quad (15)$$

where *trans*(f_i) returns the English translations of the French word f_i.

14.5 Experiments

We experiment with the task of reranking the candidate translations based on the context of a French focus word. We train all the models on approximately 20K tokens between French and English and are evaluated on approximately 7.4K tokens. Both the training and test tokens come from the EMEA domain and the translation dictionary used to train the type vectors is obtained by running giza++ on the Hansards French-English parallel data. We automatically word align the parallel data in both the directions and intersect the alignments. We also ignore all the tokens whose fertility is more than one. We only select French and English word pairs that are aligned at least 20 times, hence the French words that we consider are highly ambiguous. We report the accuracy of the top scored translation according to different models.

Method	Accuracy
Random	40.29
Max. Probable	57.84
Best cue-word	61.85
Token Adapt.	55.20
Co-regularization	59.15
Disc. Adapt.	60.21
PSD. Classifier	70.10

Table 27: Accuracy of the top-ranked translation.

Table 27 shows the results of few baseline systems and different adaptation models. The first two baselines, 'Random' and 'Max. Probable' choose a random word and maximum probable word according to $p(e_j|f_i)$ as the translation respectively. Both these models ignore the context words around the focus word. The third baseline, 'Best cue-word' uses context words. Given a focus word, first it selects a best cue-word based on the training data and choose the translation given the focus word and the best cue word. Simple token adaptation yields lower results but the two extensions co-regularization and discriminative adaptation give approximately 5 point improvement over the plain adaptation model. But the performance is still less than the best cue-word based method. Finally, we also report results using a phrase sense disambiguation (PSD) classifier. We train a classifier using context words, POS tags and positional features. The PSD classifier gives best results. But, notice that PSD classifier uses additional information that is not available to the other baselines.

14.6 Future Work

In this work, we explored the idea of adapting word type embeddings to the token level based on the context words. All the models proposed here use weighted linear combination as the model for the adaptation. In the error analysis, we observed that the training accuracies are also very low which indicates that the model is not sophisticated enough to capture the intricacies of the problem. We want to extend the adaptation model by associating a transformation matrix with each cue word rather than a scalar weight. We also want to see if providing the score computed by our model as a feature to the PSD classifier will improve the accuracies of the PSD classifier.

15 Summary and Conclusion

15.1 Summary

15.1.1 Analysis of domain effects

We conducted a detailed analysis of domain effects in SMT, and showed that they are not uniform across domains. Starting from the Canadian Hansard as the OLD domain, we consider 4 NEW domains. While moving to the News domain does not significantly benefit from NEW domain data, all other domains (Medical, Subtitles, Science) benefit substantially from NEW data.

We showed that standard simple SMT adaptation methods are only sometimes effective. Concatenating OLD and NEW data often harms translation quality in both the OLD and NEW domain. Linear and log-linear mixture models are a better starting point, but there is large room for improvement.

We showed that domain shift errors are distributed amongst 3 major categories in most NEW domains: SEEN (OOV in the NEW domain), SENSE (word that is known in the OLD domain but is translated in a previously unseen sense in the NEW domain), and SCORE (known word with known translations but different translation probability distributions in the OLD and NEW domains). This suggests that fine-grained approaches to domain adaptation that take into account global and local contextual information are necessary to substantially improve translation quality.

15.1.2 Phrase Sense Disambiguation for DAMT

In order to model context in SMT, we used "Phrase Sense Disambiguation" (PSD), which is a discriminative context-dependent translation model for SMT.

We showed that it can model lexical choice across domains. In intrinsic lexical choice tasks, we showed that sentence-level context alone can fix lexical choice errors when shifting domains. We also applied statistical domain adaptation algorithms to PSD classifiers. However, these algorithms have not proved useful yet.

Note that PSD training and scoring is fully integrated in Moses. We implemented a fast fully-automated experiment pipeline that is easy to use and extend. We are currently tracking a few remaining bugs.

15.1.3 Mining New Senses and their Translations

We have proposed promising methods to

- detect new senses. We explored a vast feature space including n-gram, topics, marginal matching, language model perplexity and others, and improved performance from mid 60s AUC to over 70 in the Medical and Science domains.
- mine useful translation for OOVs from both comparable and parallel data. In particular, we designed a new method based on document pair marginal matching, and extended the use of low-dimentional embeddings to mine translations at the token rather than type level.
- learn topic distinctions targeted at MT.

15.2 Contributions

15.2.1 Engineering Contributions

Vowpal Wabbit The work done for the workshop resulted in significant changes to the Vowpal Wabbit toolkit.

Most importantly, VW is now not only a stand-alone tool that has to be run from the command line, but also a fully linkable library that can be directly integrated in other software projects.

In addition, we contributed significant extensions to the core classifier that make it better suited to NLP applications. VW now supports (1) label-dependent features that are commonly used in NLP reranking tasks, (2) cost sensitive classification, and (3) complex feature interaction.

All these changes are publicly available from the DAMT branch of the vowpal wabbit git repository.

Moses We contributed to the Moses Statistical Machine Translation toolkit. Several optimizations were added. For instance, we parallelized significance-based phrase-table pruning to reduce the time needed to run the full training pipeline. We improved the experiment management system and fixed many bugs. DAMT team members submitted 247 commites to github and added 6917 lines of code.

The most significant contribution is the integration of VW and Moses, which represents the first integration of a general purpose classifier in Moses. It is a solid and tight integration. VW-augmented Moses can be built and run out-of-the-box. It is fully integrated in the experiment management system and can be easily extended with new features thanks to a flexible interface.

This tight integration makes VW-Moses remarkably fast: phrase-based decoding takes 180% run time of standard Moses, and is fully parallelized.

Finally, VW was integrated both in phrase-based and Hiero Moses. A common interface was designed to allow for consistent feature definitions.

15.2.2 Methodology Contributions

The workshop also contributed to defining a methodology for domain adaptation work.

We defined several MT domain adaptation tasks using the Canadian Hansard as the OLD domain, and three very different NEW domains: Medical, Science and Subtitles. We defined controlled experimental conditions to compare and contrast the impact of adaptation algorithms in these various conditions.

We also defined two stand-alone translation lexical choice tasks: (1) translation disambiguation, and (2) new translation sense detection. These tasks are defined on the exact same data as the MT test sets and target domain-relevant vocabulary. They allow to evaluate MT system components and very heterogeneous systems on the same subtasks, even before full integration in a SMT system.

We defined an experiment management system for automatic evaluation of new features (for new sense detection.)

All our data sets and evaluation frameworks will be freely available online.

15.2.3 New Techniques

Our contributions include several new techniques.

First, we conducted the first complex classifier integration into a SMT decoder, with a feature extraction framework shared between the Hiero and Phrase-based decoding frameworks.

Second, we proposed a new discriminative topic model that is domain-specific and translation aware.

Third, we introduced a new method for translation mining based on document-pair marginal matching.

Finally, we proposed an approach to perfom dictionary mining at the token levelas needed in domain-adapation settings, rather than at the coarse type level that has been more commonly explored in previous work.

15.3 Future work

Immediate next steps:

- Debug extrinsic PSD
- Improve DA representation
- Extend soft-syntactic features for Hierarchical Moses further
- Integrate mined translation examples and topic models into MT and PSD
- Package up data and software for release
- Moses+VW already available!

Longer term research directions:

- Non-lexical domain divergence issues: we have promising preliminary results using syntax
- Other language pairs and directions (More distant language; Into morphologically richer languages)

- Less structured text/genre (e.g., informal communication)
- Scale topic models to really large heterogeneous corpora: break away from OLD to NEW domain adaptation and move toward web translation

15.4 Acknowledgments

We would like to thank:

- George Foster, Colin Cherry and the Portage team @NRC
- John Langford
- Moses-support
- Cameron Macdonald, Patrik Lambert, Holger Schwenk
- Vlad Eidelman, Kristy Hollingshead, Wu Ke, Gideon Maillette de Buy Wenniger, Ferhan Ture
- Dan Povey
- Sanjeev, Monique, Ruth, Lauren, Mani*, and CLSP

We gratefully acknowledge the generous sponsors of the JHU Summer Workshop 2012, including NSF, Google and DOD. Fraser and Braune's work was partially funded by Deutsche Forschungsgemeinschaft grant SCHU 2246/6-1 "Models of Morphosyntax for Statistical Machine Translation". This work was supported in part by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886.

References

Vamshi Ambati and Alon Lavie. Improving syntax driven translation models by re-structuring divergent and non-isomorphic parse tree structures. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*, 2008.

Nadav Berman, Arie Feuer, and Elias Wahnon. Convergence analysis of smoothed stochastic gradient-type algorithm. *International Journal of Systems Science*, 18(6):1061–1078, 1987.

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

Michael Bloodgood and Chris Callison-Burch. Bucking the trend: Large-scale cost-focused active learning for statistical machine translation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 854–864, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P10-1088.

Julius R. Blum. Multidimensional stochastic approximation methods. *Ann. Math. Statistics*, 25:737–744, 1954.

Roberto Calandra. Rprop toolbox for MATLAB. http://www.ias.informatik.tu-darmstadt.de/Research/RpropToolbox, 2011.

Chris Callison-Burch and Mark Dredze. Creating speech and language data with amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10. Association for Computational Linguistics, 2010.

Marine Carpuat. One translation per discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pp. 19–27, Boulder, Colorado, June 2009. URL http://www.aclweb.org/anthology/W09-2404.

Marine Carpuat and Dekai Wu. Improving Statistical Machine Translation using Word Sense Disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pp. 61–72, Prague, June 2007.

Colin Cherry and George Foster. Batch tuning strategies for statistical machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2012.

D. Chiang. Learning to translate with source and target syntax. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1443–1452. Association for Computational Linguistics, 2010.

David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, 2005.

David Chiang, Steve DeNeefe, and Michael Pust. Two easy improvements to lexical weighting. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 455–460, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-2080.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585, December 2006.

Hal Daumé III. Frustratingly easy domain adaptation. In *Conference of the Association for Computational Linguistics (ACL)*, Prague, Czech Republic, 2007. URL http://pub.hal3.name/#daume07easyadapt.

Hal Daumé III and Jagadeesh Jagarlamudi. Domain adaptation for machine translation by mining unseen words. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, 2011.

Qing Dou and Kevin Knight. Large scale decipherment for out-of-domain machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

Vladimir Eidelman, Jordan Boyd-Graber, and Philip Resnik. Topic models for dynamic translation model adaptation. In *Association for Computational Linguistics*, 2012.

George Foster and Roland Kuhn. Mixture-model adaptation for smt. In *Proceedings of the Workshop on Statistical Machine Translation (WMT)*, 2007.

George Foster, Cyril Goutte, and Roland Kuhn. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2010.

Alexander Fraser, Marion Weller, Aoife Cahill, and Fabienne Cap. Modeling inflection and word-formation in SMT. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 664–674, Avignon, France, April 2012.

Pascale Fung and Lo Yuen Yee. An IR approach for translating new words from nonparallel, comparable texts. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, 1998.

Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers and Mathematics with Applications*, 2(1):17 – 40, 1976.

Roland Glowinski and A. Marrocco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité, d'une classe de problèmes de dirichlet non linéaires. *Rev. Franc. Automat. Inform. Rech. Operat.*, 140:41–76, 1975.

Gurobi Optimization Inc. Gurobi optimizer reference manual, 2013. URL http://www.gurobi.com.

Barry Haddow and Philipp Koehn. Analysing the effect of out-of-domain data on smt systems. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pp. 422–432, Montréal, Canada, June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W12-3154.

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

H. Hoang and P. Koehn. Improved translation with source syntax labels. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pp. 409–417. Association for Computational Linguistics, 2010.

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

Jagadeesh Jagarlamudi and Hal Daum III. Extracting multilingual topics from unaligned comparable corpora. In *Advances in Information Retrieval*, pp. 444–456, 2010. URL http://dx.doi.org/10.1007/978-3-642-12275-0_39.

R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987. ISSN 0010-485X. doi: http://dx.doi.org/10.1007/BF02278710.

Alexandre Klementiev and Dan Roth. Weakly supervised named entity transliteration and discovery from multilingual comparable corpora. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, 2006.

Philipp Koehn and Kevin Knight. Learning a translation lexicon from monolingual corpora. In *ACL Workshop on Unsupervised Lexical Acquisition*, 2002.

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

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. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, 2007.

Patrik Lambert, Holger Schwenk, and Frdric Blain. Automatic translation of scientific documents in the hal archive. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

David Lee. Genres, registers, text types, domains and styles: Clarifying the concepts and navigating a path through the bnc jungle. *Language and Computers*, 42(1):247–292, 2002.

Y. Liu, H. Mi, Y. Feng, and Q. Liu. Joint decoding with multiple translation models. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pp. 576–584. Association for Computational Linguistics, 2009.

Y. Marton and P. Resnik. Soft syntactic constraints for hierarchical phrased-based translation. *Proceedings of ACL-08: HLT*, pp. 1003–1011, 2008.

Mausam, Stephen Soderland, Oren Etzioni, Daniel S. Weld, Kobi Reiter, Michael Skinner, Marcus Sammer, and Jeff Bilmes. Panlingual lexical translation via probabilistic inference. *Artificial Intelligence*, 174:619–637, June 2010.

David Mimno, Hanna Wallach, Jason Naradowsky, David Smith, and Andrew McCallum. Polylingual topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2009.

Robert C. Moore and William Lewis. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10. Association for Computational Linguistics, 2010.

Preslav Nakov and Hwee Tou Ng. Improved statistical machine translation for resource-poor languages using related resource-rich languages. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pp. 1358–1367, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

Malte Nuhn, Arne Mauser, and Hermann Ney. Deciphering foreign language by combining language models and context vectors. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, July 2012.

Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51, March 2003. ISSN 0891-2017.

John C. Platt, Kristina Toutanova, and tau Wen Yih. Translingual document representations from discriminative projections. In *EMNLP*, pp. 251–261, 2010.

Matt Post, Chris Callison-Burch, and Miles Osborne. Constructing parallel corpora for six indian languages via crowdsourcing. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pp. 401–409, Montréal, Canada, June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W12-3152.

Emmanuel Prochasson and Pascale Fung. Rare word translation extraction from aligned comparable documents. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11. Association for Computational Linguistics, 2011.

Novi Quadrianto, Alex J. Smola, Tiberio S. Caetano, and Quoc V. Le. Estimating labels from label proportions. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pp. 776–783, New York, NY, USA, 2008. ACM.

Reinhard Rapp. Identifying word translations in non-parallel texts. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, 1995.

Reinhard Rapp. Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, 1999.

Sujith Ravi and Kevin Knight. Deciphering foreign language. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, 2011.

S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *Proceedings of the Text REtrieval Conference*, 1994.

Charles Schafer. *Translation Discovery Using Diverse Similarity Measures*. PhD thesis, Johns Hopkins University, 2006.

Charles Schafer and David Yarowsky. Inducing translation lexicons via diverse similarity measures and bridge languages. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*, 2002.

P. Simianer, S. Riezler, and C. Dyer. Joint feature selection in distributed stochastic learning for large-scale discriminative training in smt. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, 2012.

David Sontag, A. Globerson, and Tommi Jaakola. *Introduction to dual decomposition for inference*, chapter 1. MIT Press, 2010.

J.C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *Automatic Control, IEEE Transactions on*, 37(3):332 –341, mar 1992.

Jörg Tiedemann. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov (eds.), *Recent Advances in Natural Language Processing*, volume V, pp. 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria, 2009. ISBN 978 90 272 4825 1.

A. Zollmann and A. Venugopal. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation*, pp. 138–141. Association for Computational Linguistics, 2006.