A Revised and Extended Version of Latent Domain Translation Models in Mix-of-Domains Haystack

Hoang Cuong and **Khalil Sima'an**
Institute for Logic, Language and Computation
University of Amsterdam
Science Park 107, 1098 XG Amsterdam, The Netherlands

Abstract

The problem of selecting data for training machine translation (MT) systems has received attention in the past years because training MT systems on larger corpora does not always yield improved performance. The data selection problem aims at selecting training data from a mixed-domain corpus for a translation task exemplified by a small sample of in-domain data. In (Cuong and Sima'an, 2014b) we presented a new approach for selection based on a probabilistic model for data selection and adaptation. Recently it came to our attention that the online toolkit, which claims to implement (Axelrod et al., 2011), is seriously flawed which renders the baseline experiments in our paper incorrect. In this report we rectify the experimental results pertaining to the baseline approach, the method of (Axelrod et al., 2011). We also provide an in-depth analysis of the methods and show that our suggestion for exploiting the contrast between in-domain and out-domain leads to improved results for (Axelrod et al., 2011). We also report a range of experiments comparing the two approaches for data selection both on selection tasks and on translation tasks showing the relative advantages of each approach.

1 Motivation

Using larger training data is usually an effective way for improving the translation quality of SMT systems. However, this is not the case once we are working with a translation task concerning a specific domain that is narrower or different from the domain of the training corpus. The relevance of the parallel training corpus to the translation task at hand can be decisive for system performance. In practice, it has been observed that training an SMT system on a small, yet well-selected sample from a large, but mixed (and potentially noisy) parallel corpus, could produce a significantly better translation. This, however, usually requires knowing the domain of the translation task in advance, which could be represented by a small sample of in-domain data exemplifying the task.

Following in the footsteps of Axelrod et al. (2011), there is an increasing number of works that focus on improving training by data selection, e.g. (Zhang and Chiang, 2014), (Kirchhoff and Bilmes, 2014), (Duh et al., 2013), (Mansour and Ney, 2014), (Cuong and Sima'an, 2014b). Existing work can be roughly classified by two kinds of information used for selection: (1) most approaches use monolingual information as the main component for a data selection model, and (2) more recent approaches use translation model information, claimed to provide an incremental contribution to the selection performance. Thusfar, a thorough exploration of the two kinds of information, and how they contribute to selection remains unclear. This is one of the main goals of this paper.

In a recent work (Cuong and Sima'an, 2014b), we made three novel contributions:

- Introducing a novel latent variable model which lends itself to data selection but also directly to domain adaptation,

- Extending the selection model with (pseudo-) out of domain language models (LMs), i.e., instead of exploiting the contrast between the in-domain-/mixed-domains- LMs, we model the selection by exploiting the contrast between in-domain- and out-of-domain- LMs. The latter models are bootstrapped.

- Extending the selection model to exploit bilingual translation probabilities expressed over word-alignments.

Unfortunately, some time after the publication of (Cuong and Sima'an, 2014b), we became aware of two crucial bugs in an open source software[1] which we used for testing the baseline, i.e., (Axelrod et al., 2011). One of these bugs concerns the computation of the model score - which was implemented incorrectly in the baseline toolkit we used.[2] The second concerns the selection of random subsets in the toolkit. Luckily, after re-examining the method of (Cuong and Sima'an, 2014b) carefully, we find that the the main contributions of our study are still valid. In this article we thus provide additional detail with updated results, not only with correct results for the baseline, but also with extra experiments aiming for a more thorough investigation of the impact of the main contributions of (Cuong and Sima'an, 2014b).

Nevertheless, we also find that some claims in (Cuong and Sima'an, 2014b) pertaining to the baseline are an artefact of the bugs in the baseline toolkit. Contrary to the findings in our original paper, which reports that the language model contribution was minimal, after using the correct baseline, we find that both components, translation and language models, contribute significantly to the performance of the selection methods. In light of this, an additional aim of this article is to conduct a more extensive analysis of the performance of the two models, baseline and Invitation model (Cuong and Sima'an, 2014b).

2 Task Definition and Set up

Given our mixed-domains corpus C_{mix}, data selection aims at learning the most relevant sentences to a task exemplified by an in-domain seed corpus C_{in}. In the work of (Cuong and Sima'an, 2014b), we present a novel latent variable model to address the problem where we introduce a latent variable D to indicate the domain each sentence belongs to. There are simply two possible values for D in our case study, D_1 to indicate in-domain data and D_0 to indicate out-domain data; thereby presupposing that the sentences in C_{mixed} distribute over two latent corpora C_1 and C_0. In other words, we aim to assign every sentence $\mathbf{s} \in C_{mix}$ an expected count $P(D_x \mid \mathbf{s})$ that it is in $C_x \in \{C_0, C_1\}$. Eventually, we rank the corpus, select a subset that is most relevant to the task. Obviously, the performance of such a selection model lies in the way it scores sentence pairs in the mixed-domains data (C_{mix}).

Before thoroughly investigating the data selection problem, it would be useful to create some simple tasks first, with intrinsic evaluations for the selection performance. In this way, we expect to have a comprehensive study of the problem and the way standard methods address it in detail.

Selection Tasks: We use three domain-specific bilingual corpora: *Hardware* ($102,145$ sentence pairs), *Legal* ($98,667$ sentence pairs), and *Pharmacy* ($103,814$ sentence pairs). We split each of them into two separate parts - the first part will be used as the in-domain data, and the second part will be used to create our mixed-domains dataset by `hiding` it in a larger corpus. In detail, we split the Hardware part into two separate corpora with $60,000$ sentence pairs (part hidden in mixed-domain corpus) and $42,145$ sentence pairs (the in-domain part) respectively. Similarly, we split the Legal subset into two parts of $60,000$ sentence pairs (hidden part) and $38,667$ sentence pairs (the in-domain part). Finally, we split the Pharmacy subset into two parts: $60,000$ sentence pairs (hidden part) and $43,814$ sentence pairs (the in-domain part). Combining all the hidden parts together results in a mixed-domains dataset C_{mix} of $180,000$ sentences.

[1] The baseline toolkit software can be found on the link: https://github.com/lukeorland/moore_and_lewis_data_selection.

[2] Special thanks to Rico Sennrich for pointing out the important bug in the baseline toolkit we used for our previous experiments.

Intrinsic Evaluation: Following (Cuong and Sima'an, 2014b), we report the `pseudo` precision at the sentence-level using a range of cut-off criteria for selecting the top sentences in the mixed-domain corpus. With this simple task - where each domain is fairly different from each other - a good relevance model is thus expected to score higher the hidden in-domain data.

Overview Data Selection

In general, there are different ways of doing selection in the literature. The most popular approach, Axelrod et al. (2011) is to select sentence pairs in C_{mix} using the cross-entropy difference between in- and mixed-domains language models, both on source and target sides. This is a modification of the method proposed in Moore and Lewis (2010). In fact the work of Moore and Lewis (2010) was proposed to address exactly the same problem we are discussing, but for only one side (i.e. monolingual data). Thus in order to have a thorough understanding of the work of Axelrod et al. (2011), let us step back and take a look at the work of Moore and Lewis (2010) in some detail.

3 Data selection on monolingual data

Like our previous work, in this article we take a probabilistic view of data selection (and adaptation), with the aim of exposing shared and different aspects between our work and the relevant related work, particularly the work of Axelrod et al. (2011).

Given a sentence $\mathbf{s}$, let us assume a latent variable D which takes two values D_1 and D_0, standing respectively for in- and out-domain identities. Now we can model the domain choice for sentence $\mathbf{s}$ with the probability $P(D|\ \mathbf{s})$:

$$P(D|\ \mathbf{s}) = \frac{P(\mathbf{s}|\ D)P(D)}{P(\mathbf{s})} \tag{1}$$

Obviously, to rank the instances in the corpus and make a selection, we need an accurate estimate of $P(D_1|\ \mathbf{s})$.

The approach of Moore and Lewis (2010) avoids computing $P(D_1|\ \mathbf{s})$ directly, because we only need their ranking for data selection. So let us rank all sentences $\mathbf{s}$ according to some indirect score. For simplicity, let us first ignore $P(D_1)$ because the role of $P(D_1)$ is indeed minor. We now may rank simply according to $\frac{P(\mathbf{s}|\ D_1)}{P(\mathbf{s})}$. Converting this formula into the $\log$ format, we will rank according to $\log P(\mathbf{s}|\ D_1) - \log P(\mathbf{s})$. Following Moore and Lewis (2010), $P(\mathbf{s}|D_1)$ is estimated by training a language model on the (provided) in-domain data C_{in}. Whereas, $P(\mathbf{s})$ is estimated by training another language model on a (random) sample of *mixed-domains* data C_{mix}. Intuitively, the size of the random subset should be comparable to the given in-domain data.

Furthermore, to avoid the potential problem with sentence length, the cross entropy can be used rather than $\log P(\mathbf{s}|\ D_1)$, i.e. dividing $\log P(\mathbf{s}|\ D_1)$ by the length of $\mathbf{s}$ as a *normalization*. In fact, this simple normalizing technique seems to be particularly effective for the task, thereby avoiding biasing the selection to only shorter sentences. We end up with a simple but elegant formula for data selection that involves cross-entropy as follows:

$$score_{ML}(\mathbf{s}) = H(\mathbf{s}; D_1) - H(\mathbf{s}). \tag{2}$$

Summarizing the approach in Moore and Lewis (2010), Eq 2 provides a efficient solution to data selection, which tends to work reasonably well in practice. However, this method does not as well as desired for certain scenarios as we show next. We now conduct experiments with the tasks we have introduced in Section 2. Following (Moore and Lewis, 2010), we use only words that appeared more than once in the vocabulary. Moreover, we also force the vocabulary to be the same for both LMs (i.e. we build LMs from a random subset of C_{mixed} with vocabulary restricted by non-singleton tokens from the in-domain corpus. Note that to sample a subset of C_{mixed} that is similar (in terms of size) to the in-domain data, we use the toolkit of https://github.com/lukeorland/moore_and_lewis_data_selection (with

minor modification with respect to the randomization). To score the sentences, we rather rely on the code of Sida Wang (with some modification) at StanfordNLP (ModifiedMooreLewisCorpusSelection.java).[3]

Table 1 presents the result in detail, revealing the precision in selection over a variety of different cut-offs.

	Source Side			Target Side		
	True	**False**	**Pre(%)**	**True**	**False**	**Pre(%)**
Task - Pharmacy						
10000	9972	28	99.72	9957	43	99.57
20000	19867	133	99.34	19812	188	99.06
30000	29587	413	98.62	29401	599	98.00
40000	38211	1789	95.53	37595	2405	93.99
50000	43148	6852	86.30	42127	7873	84.25
60000	45605	14395	76.01	44672	15328	74.45
Task - Legal						
10000	9665	335	96.65	9650	350	96.5
20000	18468	1532	92.34	18416	1584	92.08
30000	26022	3978	86.74	25647	4353	85.49
40000	31704	8296	79.26	30898	9102	77.25
50000	35850	14150	71.70	34830	15170	69.66
60000	39090	20910	65.15	38089	21911	63.48
Task - Hardware						
10000	9787	213	97.87	9608	392	96.08
20000	17608	2392	88.04	17179	2821	85.90
30000	22454	7546	74.85	22126	7874	73.75
40000	25711	14289	64.28	25532	14468	63.83
50000	28586	21414	57.17	28476	21524	56.95
60000	31235	28765	52.06	31227	28773	52.05

Table 1: Model performance with traditional Moore-Lewis method ($score_{ML}(\mathbf{s}) = H(\mathbf{s}; D_1) - H(\mathbf{s})$).

As we can see, the method works particularly well for the case of Pharmacy,[4] and reasonably well for the case of Legal. Meanwhile for Hardware the results can be improved significantly.

Now we refine the method of Moore-Lewis, showing how it can be improved also for retrieving Hardware. In (Cuong and Sima'an, 2014b), we attempt a new approach for computing $P(\mathbf{s})$ – which we believe is the weakest point of the method of Moore-Lewis. Our claim is that the mixed-domain language models trained on a mix of rather diverse set of domains could be considered kind of wide-coverage, which makes for a rather weak contrast with the in-domain language models. Recall that the denominator can be represented as follows:

$$P(\mathbf{s}) = \sum_{D} P(\mathbf{s}|\ D)P(D) = P(\mathbf{s}|\ D_1)P(D_1) + P(\mathbf{s}|\ D_0)P(D_0) \tag{3}$$

Following this line of thought, rather than directly computing $P(\mathbf{s})$ from a random subset of C_{mix} we totally rely on both $P(\mathbf{s}|\ D_1)$ and $P(\mathbf{s}|\ D_0)$ - the LMs with respect to in-/vs out-domain. The intuition here is straightforward - a sentence that we prefer selecting should satisfy both requirements: it is *must relevant* to the in-domain data *and also irrelevant* to the out-of-domain data. In other words, instead of exploiting the contrast of sentences between the in-domain-/mixed-domains- LMs, we model the selection through exploiting the contrast between the in-domain-/out-of-domain- LMs.

Technically, while it is simple to compute $P(\mathbf{s}|\ D_1)$, it is not straightforward to compute $P(\mathbf{s}|\ D_0)$. This is because there is no out-of-domain sample that we are provided in advance. It is, however, possible to create a `pseudo` out-of-domain subset that *contrasts* most to the in-domain data. The way of creating such a subset can be done by a very simple strategy. First, we can do exactly what Moore and Lewis did, but just select a subset that is most irrelevant to the in-domain data instead. After that, we can train

[3]The original code computes perplexity difference instead of cross-entropy difference.

[4]This is not surprising, as the Pharmacy data is apparently significantly different from the rest of the data.

LMs on that subset, assuming that we have out-of-domain LMs. To avoid overfitting, we normally pick up a subset that is exactly the same size as the in-domain data. The interesting thing, however, is that normally, choosing a different size of subset could also improve the selection performance significantly.[5] We leave the question of how to pick the optimal size of out-of-domain data for future work.

To create the pseudo out-of-domain sample, let us `reuse` the ranking output from what we got with the traditional Moore-Lewis method. We then pick up the lowest ranking subset to train our out-of-domain LMs over the data. In other words, instead of using language models trained on a subset of C_{mixed} to compute $P(s)$, we rather use another language model that is trained on the pseudo out-of-domain data. Normalizing the LMs by sentence length, a new formula (yet still in the same spirit of the original formula) for the selection can be written is as follows:

$$score_{\widehat{ML}}(\mathbf{s}) = H(\mathbf{s}; D_1) - H(\mathbf{s}; D_0). \tag{4}$$

Here, we use notation $score_{\widehat{ML}}(\mathbf{s})$ to denote the new way of doing data selection yet still in the spirit of the work of (Moore and Lewis, 2010). By using $score_{\widehat{ML}}(\mathbf{s})$ instead of $score_{ML}(\mathbf{s})$, we expect the contrast between in-/out- would produce a better fitting to the data selection problem.

	Iteration 1					
	Source Side			**Target Side**		
	True	**False**	**Pre(%)**	**True**	**False**	**Pre(%)**
Task - Pharmacy						
10000	9999	1	99.99	9999	1	99.99
20000	19993	7	99.97	19997	3	99.99
30000	29970	30	99.90	29971	29	99.90
40000	39868	132	99.67	39854	146	99.64
50000	49065	935	98.13	48928	1072	97.86
60000	52809	7191	88.02	52666	7334	87.78
Task - Legal						
10000	9951	49	99.51	9958	42	99.58
20000	19748	252	98.74	19792	208	98.96
30000	29142	858	97.14	29209	791	97.36
40000	37449	2551	93.62	37459	2541	93.65
50000	43224	6776	86.45	43338	6662	86.68
60000	46759	13241	77.93	46884	13116	78.14
Task - Hardware						
10000	9931	69	99.31	9905	95	99.05
20000	18866	1134	94.33	18911	1089	94.56
30000	25763	4237	85.88	25596	4404	85.32
40000	30867	9133	77.17	30324	9676	75.81
50000	34438	15562	68.88	33950	16050	67.90
60000	36968	23032	61.61	36668	23332	61.11

Table 2: Model performance with the refinement in the rank computation ($score_{\widehat{ML}}(\mathbf{s}) = H(\mathbf{s}; D_1) - H(\mathbf{s}; D_0)$) - Iteration 1.

Table 2 presents the result, revealing an improvement via better precision in selection over a variety of different cut-offs. That is, using out-of-domain LMs instead of mixed-domain LMs, we observe a consistent improvement in the selection performance over three tasks of Legal, Hardware and Pharmacy.

The idea of pseudo out-domain language model seems to work. To confirm the result, we now reuse the ranking output from $score_{\widehat{ML}}(\mathbf{s})$ (call this `iteration 1`). We then set aside the `most contrasting` subset to the in-domain data, which are the lowest ranked sentences, and train our out-of-domain LMs over this data, and repeat the experiments. Table 3 presents the result in detail, revealing slightly better model performance compared to what we obtained at iteration 1.

Finally, we try another iteration of the training to investigate the model performance. As expected, the performance is slightly better than the outcome of iteration 2. Table 4 shows the result in detail.

[5]However, note that in certain cases this could also hurt the selection performance significantly.

Iteration 2						
	Source Side			Target Side		
	True	False	Pre(%)	True	False	Pre(%)
Task - Pharmacy						
10000	9999	1	99.99	9999	1	99.99
20000	19994	6	99.97	19997	3	99.99
30000	29978	22	99.93	29980	20	99.93
40000	39895	105	99.74	39877	123	99.69
50000	49232	768	98.46	49100	900	98.2
60000	53468	6532	89.11	53236	6764	88.72
Task - Legal						
10000	9951	49	99.51	9958	42	99.58
20000	19748	252	98.74	19792	208	98.96
30000	29142	858	97.14	29209	791	97.36
40000	37449	2551	93.62	37459	2541	93.65
50000	43224	6776	86.45	43338	6662	86.68
60000	46759	13241	77.93	46884	13116	78.14
Task - Hardware						
10000	9931	69	99.31	9905	95	99.05
20000	18866	1134	94.33	18911	1089	94.56
30000	25763	4237	85.88	25596	4404	85.32
40000	30867	9133	77.17	30324	9676	75.81
50000	34438	15562	68.88	33950	16050	67.90
60000	36968	23032	61.61	36668	23332	61.11

Table 3: Model performance with the refinement in the rank computation ($score_{\widehat{ML}}(\mathbf{s}) = H(\mathbf{s}; D_1) - H(\mathbf{s}; D_0)$) - Iteration 2.

Iteration 3						
	Source Side			Target Side		
	True	False	Pre(%)	True	False	Pre(%)
Task - Pharmacy						
10000	9999	1	99.99	9999	1	99.99
20000	19994	6	99.97	19997	3	99.99
30000	29980	20	99.93	29975	25	99.92
40000	39893	107	99.73	39875	125	99.69
50000	49266	734	98.53	49133	867	98.27
60000	53572	6428	89.29	53352	6648	88.92
Task - Legal						
10000	9965	35	99.65	9969	31	99.69
20000	19813	187	99.07	19835	165	99.18
30000	29354	646	97.85	29407	593	98.02
40000	37885	2115	94.71	37949	2051	94.87
50000	44087	5913	88.18	44281	5719	88.56
60000	47897	12103	79.83	48016	11984	80.03
Task - Hardware						
10000	9940	60	99.40	9774	226	97.74
20000	18834	1166	94.17	18855	1145	94.28
30000	26121	3879	87.07	25785	4215	85.95
40000	31525	8475	78.81	30848	9152	77.12
50000	35440	14560	70.88	34712	15288	69.42
60000	38183	21817	63.64	37664	22336	62.77

Table 4: Model performance with the refinement in the rank computation ($score_{\widehat{ML}}(\mathbf{s}) = H(\mathbf{s}; D_1) - H(\mathbf{s}; D_0)$) - Iteration 3.

However, the introduction of LMs over pseudo out-of-domain (D_0) in our work is one aspect of a broader picture which we go through in the following sections.

4 Data selection on bilingual data

So far we have presented the traditional way of data selection on monolingual data. We have presented the classic work of Moore and Lewis, analysing some of its properties. We also proposed a different

approach to train the models and provided a comparison with the original approach, revealing a better performance for our proposed method - exploiting the contrast between the in-domain-LMs and pseudo out-of-domain-LMs.

Now let us come back to the main topic of this discussion - bilingual data selection. Again, given a mixed-domains corpus C_{mix}, our goal is thus to determine for every pair of parallel sentences $\{\mathbf{f},\ \mathbf{e}\}$ in C_{mix}, whether it should be in C_1 (in-domain) or C_0 (not in in-domain). Again we put a latent variable D to indicate the domain each sentence pair should be in; again D has only two values, D_1 to indicate in-domain data and D_0 to indicate out-domain data.

As a straightforward extension to the method of Moore-Lewis, we like to view Axelrod et al. (2011) as a version which aims to exploit two translation directions. Although the method is not expressed as a probability, we will here start out from a probabilistic expression and then transform it to cross entropy difference as it shows the shared and different aspects from our own work.

$$P(D|\mathbf{f},\mathbf{e}) \approx \frac{P(\mathbf{f}|D)P(\mathbf{e}|D)P(D)}{P(\mathbf{f})P(\mathbf{e})}. \tag{5}$$

Straightforwardly, for the method of Axelrod et al. (2011) we do not need to compute $P(D_1|\ \mathbf{f},\mathbf{e})$ directly, because we only need the ranking. By ignoring $P(D)$, together with converting this formula into the $\log$ format, we thus rank according to $\log P(\mathbf{f}|\ D_1) + \log P(\mathbf{e}|\ D_1) - \log P(\mathbf{f}) - \log P(\mathbf{e})$. Applying the normalization trick, we end up with a formula for data selection as follows:

$$score_{Axel}(\mathbf{f},\ \mathbf{e}) = H(\mathbf{f};\ D_1) - H(\mathbf{f}) + H(\mathbf{e};\ D_1) - H(\mathbf{e}). \tag{6}$$

As discussed above, it is expected that better selection performance can be achieved by exploiting the sharper contrast between the in-domain-/out-of-domain- LMs as follows:

$$score_{\widehat{Axel}}(\mathbf{f},\ \mathbf{e}) = H(\mathbf{f};\ D_1) - H(\mathbf{f};\ D_0) + H(\mathbf{e};\ D_1) - H(\mathbf{e};\ D_0). \tag{7}$$

As before, here we use notation $score_{\widehat{Axel}}(\mathbf{f},\ \mathbf{e})$ to denote our own modified approach of (Axelrod et al., 2011), where we exploit the contrast between in-domain and out-domain, instead of mix-domain data.

	$score_{Axel}(\mathbf{f},\ \mathbf{e})$			$score_{\widehat{Axel}}(\mathbf{f},\ \mathbf{e})$								
				Iteration 1			Iteration 2			Iteration 3		
	True	**False**	**Pre(%)**	**True**	**False**	**Pre(%)**	**True**	**False**	**Pre(%)**	**True**	**False**	**Pre(%)**
Task - Pharmacy												
10000	9985	15	99.85	9999	1	99.99	9999	1	99.99	9999	1	99.99
20000	19935	65	99.68	19998	2	99.99	19998	2	99.99	19998	2	99.99
30000	29757	243	99.19	29988	12	99.96	29991	9	99.97	29992	8	99.97
40000	38772	1228	96.93	39945	55	99.86	39955	45	99.89	39954	46	99.89
50000	43870	6130	87.74	49481	519	98.96	49571	429	99.14	49588	412	99.18
60000	46417	13583	77.36	53241	6759	88.74	53828	6172	89.71	53919	6081	89.87
Task - Legal												
10000	9809	191	98.09	9979	21	99.79	9990	10	99.9	9990	10	99.9
20000	19017	983	95.09	19879	121	99.40	19903	97	99.52	19904	96	99.52
30000	26880	3120	89.60	29493	507	98.31	29624	376	98.75	29629	371	98.76
40000	32759	7241	81.80	38231	1769	95.58	38568	1432	96.42	38645	1355	96.61
50000	36966	13034	73.93	44481	5519	88.96	45144	4856	90.29	45327	4673	90.65
60000	40079	19921	66.79	47946	12054	79.91	48815	11185	81.36	49041	10959	81.74
Task - Hardware												
10000	9847	153	98.47	9962	38	99.62	9835	165	98.35	9835	165	98.35
20000	17959	2041	89.80	19280	720	96.40	19126	874	95.63	19182	818	95.91
30000	22967	7033	76.56	26274	3726	87.58	26580	3420	88.60	26577	3423	88.59
40000	26467	13533	66.17	31401	8599	78.50	31946	8054	79.87	32076	7924	80.19
50000	29436	20564	58.88	34994	15006	69.99	35740	14260	71.48	35939	14061	71.88
60000	32115	27885	53.53	37543	22457	62.57	38488	21512	64.14	38718	21282	64.53

Table 5: Model performance with bilingual information.

Table 5 presents the results in detail. As we can see - combining the cross entropy from both sides consistently improves the selection performance. In practice, we consistently observed this, even though sometimes, it might slightly hurt. As a side note, it is not surprising to see a consistent improvement by exploiting the sharp contrast between the in-domain-/out-of-domain- LMs.

Importantly, the nice thing about the approach of cross-entry method is that it is fairly straightforward, and particularly efficient.

So far everything is okay. Applying the method of Axelrod et al. (2011), we have extended from monolingual information to bilingual information, and it works quite well. Meanwhile, our proposal to exploit the sharp contrast between the in-domain-/out-of-domain- LMs provides better performance. Nonetheless, it would be worthy to try to directly model $P(D|\mathbf{f},\ \mathbf{e})$ instead of redirecting the problem into some indirect scores. Our work of (Cuong and Sima'an, 2014b) contributes a principled way of doing so which allows not only data selection but also adaptation of a variety of models cf. (Cuong and Sima'an, 2014a; Cuong and Sima'an, 2015).

While working with bilingual data, we observe that so far in the approach of (Axelrod et al., 2011), there is no information about translation probabilities for modeling data selection. In (Cuong and Sima'an, 2014b), we claim that this is particularly useful for selection. Following the line of thought, instead of relying only on monolingual information we thus rely on a combination between monolingual information and translation information for the selection. By formalizing the problem with latent variable model, let us continue to have a full expansion of $P(D|\mathbf{f},\mathbf{e})$ as follows:

$$\begin{aligned} P(D|\mathbf{f},\mathbf{e}) &= \frac{1}{Z_D}\Big(P(\mathbf{f},\mathbf{e}|D)P(D)\Big) = \frac{1}{Z_D}\Big(\frac{1}{2}(P(\mathbf{f},\mathbf{e}|D) + P(\mathbf{f},\mathbf{e}|D))P(D)\Big) \\ &= \frac{1}{Z_D}\Big(P(D)(P(\mathbf{f}|\ \mathbf{e},\ D)P(\mathbf{e}|D) + P(\mathbf{e}|\ \mathbf{f},D)P(\mathbf{f}|\ D))\Big). \end{aligned} \tag{8}$$

Different from the traditional approach, our goal is to directly model $P(D|\mathbf{f},\mathbf{e})$, before using the result for ranking the relevance of sentences. Note that we effectively average between the two translation directions which is reasonable, as there is no reason to give preference to any of them. Here, Z_D serves as a normalization constant.

As the equation shows, the selection now is modeled by a combination between translation models and language models, where the models are with respect to in-domain and out-domain (not mixed-domain). Now let us discuss the new piece of information for the selection here - the domain-dependent translation models $P(\mathbf{f}|\ \mathbf{e},D)$, which can be viewed as modeling the probability that source/target sentence translates as target/source sentence in domain $D\ \in\ \{D_0, D_1\}$. Given $\mathbf{f} = f_1, f_2, \ldots, f_m$ and $\mathbf{e} = e_1, e_2, \ldots, e_l$, we assume hidden alignments $\mathbf{A} = a_1, a_2, \ldots, a_m$ akin to IBM Model I, i.e. $P(\mathbf{f},\mathbf{A}\ |\ \mathbf{e},D) = \frac{\epsilon}{(l+1)^m}\prod_{j=1}^{m} t(f_j|e_{a_j},D)$. Here, $t(f_j|e_{a_j},D)$ can be thought of as the domain-dependent lexical probability of f_j given e_{a_j} with respect to D. Thanks to the tractability of IBM Model I, it is straightforward to compute the translation model with respect to $P(\mathbf{f}\ |\ \mathbf{e},D)$, $P(\mathbf{f}|\ \mathbf{e},D) = \sum_{\mathbf{A}} P(\mathbf{f},\mathbf{A}|\mathbf{e},D) = \frac{\epsilon}{(l+1)^m}\prod_{j=1}^{m}\sum_{i=0}^{l} t(f_j|e_i,D)$.

Of course, one can use only the Viterbi alignment to compute $P(\mathbf{f}|\ \mathbf{e},D)$ (a.k.a. the so-called maximum-approximation), but in practice, we found that marginalizing all alignments produces much better result. As a side note, one could also use phrase-based models instead of word-based models to model $P(\mathbf{f}|\ \mathbf{e},D)$. A systematic comparison between these different ways in the model performance, however, is beyond this article.

We now turn to discuss how to train our model. The parameters Θ_D consist of the domain-dependent lexical parameters (e.g., $t_{\Theta_D}(f|e,D)$, $t_{\Theta_D}(e|f,D)$) and the domain prior parameter (e.g., $P_{\Theta_D}(D)$). With these sharpened translation and language models, training (via maximizing the likelihood) commences using a version of EM (Dempster et al., 1977). Metaphorically, iterative EM training resembles `party invitations` on social networks (hence, the Invitation model): if initially in/out-domain

sentence pairs (the hosts) invite some sentence pairs from $\mathbf{C}_{mix}$, in the next iteration the new pseudo in/out-domain sentences help invite more sentence pairs.

As a technical note - since we are working with different language models, it is important to scale the probabilities of the four LMs to make them comparable. In practice, a way which is quite simple is as follows: we just need to normalize the probability that a LM assigns to a sentence by the total probability this LM assigns to all sentences in $\mathbf{C}_{mix}$. This, however, could downplay the role of LMs in the selection model, raising a question of finding a better way for the normalization.[6]

There are two different ways of creating the pseudo out-of-domain data. The first way is to rely on the information from LMs as we did above. The second way is to rely on the information from translation models. Since we already implemented the first approach for the above experiments, let us continue following experiments with the second approach of creating such pseudo out-of-domain data. In practice, we tried to make a systematic comparison between the two approaches. Our result suggests there is not much significant difference in the performance between these different techniques.

	$score_{Axel}(\mathbf{f}, \mathbf{e})$ Axelrod et al. (2011)			**Modeling $P(D\|\mathbf{f},\mathbf{e})$ directly with monolingual and bilingual information**								
				Iteration 1			**Iteration 2**			**Iteration 3**		
	True	**False**	**Pre(%)**	**True**	**False**	**Pre(%)**	**True**	**False**	**Pre(%)**	**True**	**False**	**Pre(%)**
Task - Pharmacy												
10000	9985	15	99.85	9985	15	99.85	9974	26	99.74	9968	32	99.68
20000	19935	65	99.68	19805	195	99.02	19785	215	98.93	19759	241	98.79
30000	29757	243	99.19	28508	1492	95.03	28447	1553	94.82	28447	1553	94.82
40000	38772	1228	96.93	36732	3268	91.83	36624	3376	91.56	36483	3517	91.21
50000	43870	6130	87.74	43411	6589	86.82	42931	7069	85.86	42660	7340	85.32
60000	46417	13583	77.36	48084	11916	80.14	47737	12263	79.56	47426	12574	79.04
Task - Legal												
10000	9809	191	98.09	9791	209	97.91	9792	208	97.92	9787	213	97.87
20000	19017	983	95.09	18683	1317	93.42	18601	1399	93.00	18496	1504	92.48
30000	26880	3120	89.60	25959	4041	86.53	25872	4128	86.24	25877	4123	86.26
40000	32759	7241	81.80	32429	7571	81.07	32077	7923	80.19	31885	8115	79.71
50000	36966	13034	73.93	37552	12448	75.10	37140	12860	74.28	36829	13171	73.66
60000	40079	19921	66.79	41518	18482	69.20	41084	18916	68.47	40777	19223	67.96
Task - Hardware												
10000	9847	153	98.47	9781	219	97.81	9777	223	97.77	9771	229	97.71
20000	17959	2041	89.80	18987	1013	94.94	19122	878	95.61	19169	831	95.85
30000	22967	7033	76.56	27358	2642	91.19	27467	2533	91.56	27556	2444	91.85
40000	26467	13533	66.17	34765	5235	86.91	34978	5022	87.44	35085	4915	87.71
50000	29436	20564	58.88	40980	9020	81.96	40976	9024	81.95	40985	9015	81.97
60000	32115	27885	53.53	45658	14342	76.09	45743	14257	76.23	45693	14307	76.16

Table 6: Model performance with bilingual information, modeling $P(D|\mathbf{f},\mathbf{e})$ directly with monolingual and bilingual information.

We now run some experiments with the extension of the selection by the integration of translation information. Table 6 presents the result in detail. We make following observations:

- In general, Table 6 suggests that directly modeling $P(D|\mathbf{f},\mathbf{e})$ for the selection could provide a promising performance. In some cases, for example, the task of retrieving Hardware - we obtain a significant improvement with our aproach compared to the method of Axelrod et al. (2011).

- However, there is not always improvement and the results could be mixed. For instance, for the task of retrieving Pharmacy and Legal - we observe a worse selection performance in some cases. This suggests that we could still have improvement by some careful rethinking of the methods of selection the pseudo out-domain, scaling the LMs and balancing LMs against TMs.

- The integration of translation information into the selection provides an incremental contribution into the selection performance. However, surprisingly, the contribution of such information in

[6]In data selection, we find that a good normalization technique is indeed quite important for final performance.

general is not as effective as exploiting the contrast between the in-domain-/out-of-domain- LMs. This suggests that monolingual information and bilingual information both provide significantly valuable information for the selection, instead of only translation information as we observed from our (wrong) previous experiment results reported in (Cuong and Sima'an, 2014b).

- The pseudo-precision between each iterations is mixed - i.e. it might be increasing (implying better performance) after an EM iteration in some cases, yet it might not be the case in other cases. This confirms a certain mismatch between the likelihood and the invitation idea in training. This demands more attention in future work.

5 Translation Experiments

We now focus on extrinsic evaluation through translation experiments, revealing how these different models can help select useful subsets for training statistical machine translation. Our data for these translation experiments is from (Cuong and Sima'an, 2014b). We build an English-Spanish mixed-domain corpus consisting of a large and rather varied set of domains (a haystack) in a way that allows us to directly measure selection quality. Starting out from a mixed-domain corpus $\mathcal{C}_{mixed}$ consisting of $4.51M$ sentence pairs, collected from multiple resources including EuroParl (Koehn, 2005), Common Crawl Corpus, UN Corpus, News Commentary, TAUS Software, TAUS Hardware, and TAUS Pharmacy, and a 100K in-domain (TAUS Legal) sentence pairs. Given a corpus of 77K in-domain pairs constitute $\mathcal{C}_{in}$. We aim to see how the selection can help building small-scale SMT system over the output. Table 7 summarizes the data and the translation task.

Task	**Corpora**	**English**	**Spanish**
	Mixed-Domain Corpus (4.51M sents)	$125,339,057$	$139,655,311$
	In-Domain Corpus (77K sents)	$1,555,342$	$1,733,370$
TAUS Legal	Dev (2K sents)	$27,983$	$30,501$
	Test (2K sents)	$45,736$	$48,999$

Table 7: The data preparation - training, dev and testing corpora (size in words). Note that the dev set contains sentences of 10-25 words, while the test set contains sentences that vary substantially in length, from 5-10 words up to 45-50 words.

We deploy three frameworks for the selection of data for this translation task. The first one is the work of (Axelrod et al., 2011). The second is our refinement of (Axelrod et al., 2011), modeling the selection through exploiting the contrast between the in-domain-/out-of-domain- LMs. Finally, we directly model the probability of domain given sentences. In the end, we get the output and rank the sentence according to the result.

We use Moses (Koehn et al., 2007) with GIZA++ (Och and Ney, 2003) and k-best batch MIRA (Cherry and Foster, 2012). The system includes MOSES (Koehn et al., 2007) baseline feature functions, plus eight hierarchical lexicalized reordering model feature functions (Galley and Manning, 2008). The training data is first word-aligned using GIZA++ (Och and Ney, 2003) and then symmetrized with *grow(-diag)-final-and* (Koehn et al., 2003). We limit the phrase length to the maximum of seven words. Final MT systems use the same *non-adapted language models* trained on 2.2M English Europarl sentences plus 248.8K sentences from News Commentary Corpus (WMT 2013).

We report BLEU (Papineni et al., 2002). Statistical significance uses 95% confidence intervals using paired bootstrap re-sampling (Press et al., 1992; Koehn, 2004). The k-best batch MIRA optimizer (Cherry and Foster, 2012) was run at least three times to optimize any SMT system to avoid instability (Clark et al., 2011). Note that metric scores for the systems are averages over multiple runs.

Table 8 presents the result in detail. The result is mixed, and there is no clear winner. In other words, each method gains its best performance in a very specific case. In detail, we make the following observations:

	Axelrod et al. (2011) $score_{Axel}(\mathbf{f}, \mathbf{e})$			Cuong and Sima'an (2014b) $score_{\widehat{Axel}}(\mathbf{f}, \mathbf{e})$			Cuong and Sima'an (2014b) **Modeling** $P(D\|\mathbf{f}, \mathbf{e})$ **directly**		
	BLEU	**METEOR**	**TER**	**BLEU**	**METEOR**	**TER**	**BLEU**	**METEOR**	**TER**
Task - Legal									
50000	34.1	34.5	49.8	**36.0**	35.9	**47.8**	35.4	**36.3**	48.2
100000	36.8	36.7	47.1	**37.4**	**36.9**	**46.3**	35.9	36.7	47.3
150000	**38.1**	37.3	46.4	37.9	**37.4**	**45.8**	36.4	37.0	47.0

Table 8: Model performance with bilingual information.

- When choosing the cut-offs of 50K: it is better (in terms of BLEU, METEOR and TER) for an MT system to be trained on the output produced by exploiting the contrast between the in-domain-/out-of-domain- LMs, or modeling directly $P(D|\mathbf{f}, \mathbf{e})$.
- When choosing the cut-offs of 100K: it is better (in terms of BLEU, METEOR and TER) for an MT system to be trained on the output produced by exploiting the contrast between the in-domain-/out-of-domain- LMs.
- When choosing the cut-offs of 150K: it is better (in terms of BLEU) for an MT system to be trained on the output produced by the work of Axelrod et al. (2011). In terms of METEOR and TER, however, it would be better to build an MT system that is trained on the output produced by exploiting the contrast between the in-domain-/out-of-domain- LMs.

6 Conclusions

We finally summarize the findings from this brief study:

- The precision/recall of the task of retrieving in-domain sentences hidden in a large, mixed-domain corpus shows that our bilingual approach, which exploits translation probabilities beside language model probabilities, may give improved performance relative to existing approaches. We have seen that our variant on the approach of (Axelrod et al., 2011), which proposes to train on in-domain and out-domain (instead of in-domain and mixed-domain), gives significant improvements.
- Similarly, exploiting translation probabilities as proposed in (Cuong and Sima'an, 2014b) gives improves selection precision and recall.
- However, we also found that this improvement in selection performance using our approach (Cuong and Sima'an, 2014b) does not always materialize as improvement in translation quality (e.g. BLEU scores). In short, we have seen that our approach provides comparable translation performance to the work of (Axelrod et al., 2011), while improving selection by sharpening the contrast between in-domain and out-domain (instead of in-domain and mixed-domain).

One potential advantage for the approach proposed in (Cuong and Sima'an, 2014b) is that it can be applied to data selection as well as a range of other adaptation tasks with minor modification. The work presented in (Cuong and Sima'an, 2014a) shows how the approach can be applied to adapt all translation components of a standard phrase-based system to an in-domain task, and the work in (Cuong and Sima'an, 2015) shows how it can be applied to adapting word alignments from mixed-domain to in-domain data with significant advantages over existing approaches. The main challenge of the method, however, lies in its training, as it is apparently much harder to design a learning algorithm for training the model. Our ongoing work aims to address this challenge.

Acknowledgements

Our special thanks goes to Rico Sennrich for pointing out important bugs in the baseline toolkit we used for our previous experiments.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of EMNLP*.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the NAACL-HLT*.

Jonathan Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of ACL (Short Papers)*.

Hoang Cuong and Khalil Sima'an. 2014a. Latent domain phrase-based models for adaptation. In *Proceedings of EMNLP*.

Hoang Cuong and Khalil Sima'an. 2014b. Latent domain translation models in mix-of-domains haystack. In *Proceedings of COLING*.

Hoang Cuong and Khalil Sima'an. 2015. Latent domain word alignment for heterogeneous corpora. In *Proceedings of NAACL-HLT*.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the ACL*.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of EMNLP*.

Katrin Kirchhoff and Jeff Bilmes. 2014. Submodularity for data selection in machine translation. In *Proceedings of EMNLP*.

Philipp Koehn, Franz Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL*.

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

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of MT-Summit*.

Saab Mansour and Hermann Ney. 2014. Unsupervised adaptation for statistical machine translation. In *Proceedings of WMT*.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 220–224, Stroudsburg, PA, USA. Association for Computational Linguistics.

Franz Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, pages 19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of ACL*.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 1992. *Numerical Recipes in C: The Art of Scientific Computing*.

Hui Zhang and David Chiang. 2014. Kneser-ney smoothing on expected counts. In *Proceedings of ACL*.