Incremental Re-training for Post-editing SMT

Daniel Hardt
Computational Linguistics
Copenhagen Business School
dh.isv@cbs.dk

Jakob Elming
Computational Linguistics
Copenhagen Business School
jel.isv@cbs.dk

Abstract

A method is presented for incremental re-training of an SMT system, in which a local phrase table is created and incrementally updated as a file is translated and post-edited. It is shown that translation data from within the same file has higher value than other domain-specific data. In two technical domains, within-file data increases BLEU score by several full points. Furthermore, a strong recency effect is documented; nearby data within the file has greater value than more distant data. It is also shown that the value of translation data is strongly correlated with a metric defined over new occurrences of n-grams. Finally, it is argued that the incremental re-training prototype could serve as the basis for a practical system which could be interactively updated in real time in a post-editing setting. Based on the results here, such an interactive system has the potential to dramatically improve translation quality.

1 Introduction

While statistical machine translation (SMT) systems have been steadily improving in quality, many users find it necessary to post-edit the output. In this paper we suggest that post-edited translation data might be of great value to an SMT system, and we describe a method to take advantage of it. This is a challenge, since the standard approach to re-training a typical SMT system takes many hours or even days. In this paper we present a system that performs incremental re-training of an SMT system. For modest sized systems, re-training can be performed in a few seconds. While this is too slow for a practical interactive system, we believe that the approach described here can provide the basis for development of a practical interactive system with incremental re-training.

In this paper we use the incremental re-training system to *simulate* a post-editing situation, by translating files that are supplied with reference translations. After translating a given source sentence from the test file, we update the translation table with the source sentence together with its reference translation. The system uses the updated translation table in translating the subsequent sentence.

We are not aware of any methods proposed for incremental re-training in an interactive setting. Indeed, one might suspect that there is little to gain from this – after all, production-quality SMT systems tend to be based on many millions of words of translation data. One might expect little or no tangible effect on quality when incorporating the data from an typical interactive session, in which a paltry few hundred new sentences might be produced.

The main result of this paper is that there is such a *file-context effect*: a small amount of translation data from within a file has a striking effect on translation quality. The incremental re-training system makes it possible for us to search for file-context effects, and by demonstrating the existence of such effects, we provide motivation for developing a practical version of this method.

A natural next question to ask concerns position within a file: given that it matters whether data is from the same file or not, does it matter *where* in the file the data is found? Again, our experiments provide a clear positive answer, with a demonstration of

strong within-file recency effects.

We pose a further question: *why* does some particular translation data have the value it does? In investigating this question, we find that the value of translation data for a given test file is strongly correlated with a metric that looks at how many new n-gram matches there are. This correlation is suggestive of the possibility that one might be able to automatically assess the value of some translation data for a given purpose.

From a linguistic perspective it is perhaps not surprising that we have found strong file-context effects. After all, linguists have long recognized that there is a structure to the sentences making up a discourse, much as there is a structure to the words making up those sentences. Some reflections of such inter-sentential relations have been seen in empirically-based work in NLP: for example (Gale et al., 1992) observed a strong tendency for "one sense per discourse"; (Hearst, 1997) shows that discourse structure is strongly associated with word frequencies. However, such inter-sentential relationships have been conspicuously absent from SMT research – while increasingly effective methods are being developed for dealing with relations between words and phrases within structure of sentences, relationships between sentences are completely ignored. One way of viewing the work reported here is that it proposes some simple ways to begin to exploit the value of relationships beyond the sentence in SMT.

In what follows, we begin by describing incremental re-training, an approach where translation data is incorporated sentence-by-sentence as a file is being translated and post-edited. We describe a prototype which can rapidly approximate alignments of new translation data, and use these alignments to produce an extended phrase table. We then report on Experiment 1, concerning file-context effects – that is, we examine the value of translation data within the same file, compared to other translation data. Next, Experiment 2 concerns recency effects on the value of translation data. We then look for an explanation of the differences in value of different translation data, and we show that the value of translation data is strongly correlated with what we call *novel repeat n-grams*: n-grams that are repeated within a file, but are novel in that they do not appear in the baseline data. Finally, we examine related work and discuss ways to extend the work described here: in particular, we describe some straightforward improvements that we believe would make the incremental re-training fast enough for interactive deployment.

2 Incremental Re-Training

Consider an interactive MT post-editing setting, in which a user creates a translation by automatically translating a file one sentence at a time, and editing the current sentence before going on to the next sentence. We seek to create a system that can take advantage of this newly created translation data when translating subsequent sentences. This is a challenge, since the standard training approach requires many hours or even days for systems of reasonable size. To be practical, incremental retraining must be performed in less than one second.

Typically, the most time-consuming step in SMT training is word alignment (as for example, in the tools associated with the Moses system (Koehn et al., 2007)). We describe a simple technique to quickly approximate word alignments of a newly translated sentence. Once the alignments are determined, they can be quickly scored and incorporated into an updated phrase table. We perform experiments with *Oracle Alignments* as well as our approximate alignments.

2.1 Oracle Alignments

In this approach we perform a standard GIZA++ (Och and Ney, 2003) alignment for the input test file and its reference translation. Then after translating each sentence, we use these *Oracle Alignments* as the basis for updating the phrase table. This gives us a useful standard by which to judge the approximate alignments produced incrementally, by the approximate method described below.

2.2 Approximate Alignments

Here we describe a technique for aligning new translation data without invoking the standard, time consuming EM algorithm of GIZA++.

2.2.1 Modified GIZA++

In order to create a word alignment for novel sentence pairs, we use the final parameters produced by

GIZA++ in the training of the baseline SMT system. A gready search algorithm is employed to find the locally optimal word alignment. First, a bootstrap word alignment with one-to-one links between same sentence positions is created. This alignment is then modified iteratively until no modification leads to a higher probability for the entire word alignment. Each modification step consists of attempting to change the link of every target position to a new source positions, since the IBM models restrict each target position to carry only one link. Trying all source positions, the single link change that produces the highest probability increase of the entire word alignment according to the GIZA++ model 4 parameters is kept.

2.2.2 Post-Processing

The above application of GIZA++ is a crude approximation of the Oracle alignments that would be produced by a complete training. We improve these approximate alignments with two simple post-processing steps.

- *Find Unknowns:* the above alignment procedure is frequently confronted with previously unknown words, and these are often left unaligned. A simple heuristic solves many of these missing alignments: if the source contains an unknown word that has not been aligned, it is aligned to the first non-aligned unknown in the target.

- *Fill Holes:* the alignments leave many "holes" – positions that are not aligned in source or target. If an unaligned pair of positions is preceded and followed by corresponding alignments, the pair is added to the alignments. For example, if we have the following alignment: *1-1 2-2 4-4*, we would add the alignment *3-3*.

2.3 Local Phrase Table

Once the alignments are produced, we use them to produce an additional phrase table, using the standard training script provided with Moses: *Extract Phrases (step 5)* and *Score Phrases (step 6)*. Another alternative would be to build a single phrase table, and combine the newly aligned data with the baseline data. We believe this could also be implemented efficiently, but chose here to build a second phrase table for reasons of simplicity. In addition, we are interested in the possibility of giving a higher priority to translation data from the same file. Having a separate phrase table for this local translation data will make it simpler to experiment with such prioritization.

2.4 Decoding

Ultimately, we envision our incremental re-training process being integrated into an interactive post-editing system, exploiting the translation data as it is being produced. In the present work, we *simulate* such a setting, in the following way (see Figure 1). The system has an initial baseline setup, including language model, phrase table, and parameters obtained by a standard training process using the baseline data.

Then a testfile is translated line by line, with the reference translation playing the role of the human post-editor. This proceeds as follows: first, the system translates a line of the file. Then, that line is paired with the corresponding line from the reference translation. This pair of lines is used to update the local phrase table, using incremental re-training: heuristic or oracle word alignments are obtained, whereupon phrases are extracted, scored, and added to the local phrase table.

3 Experiment 1: File-Context Effects

The incremental re-training system points towards the possibility of a system in which post-edited SMT output sentences would be made available to the SMT system as it translates subsequent sentences within the same file. We attempt to assess the value of this data by comparing to baseline systems that do not have access to within-file translation data.

3.1 Background: Data

The experiments involved three different sets of translation data: the Danish-English portion of the Europarl corpus (Koehn, 2005), a 31 million word collection of chemical Patent translations, and a 2.1 million word collection of Clinical Trial Protocols. All the data is Danish and English, and all the experiments involved translation from Danish to English.

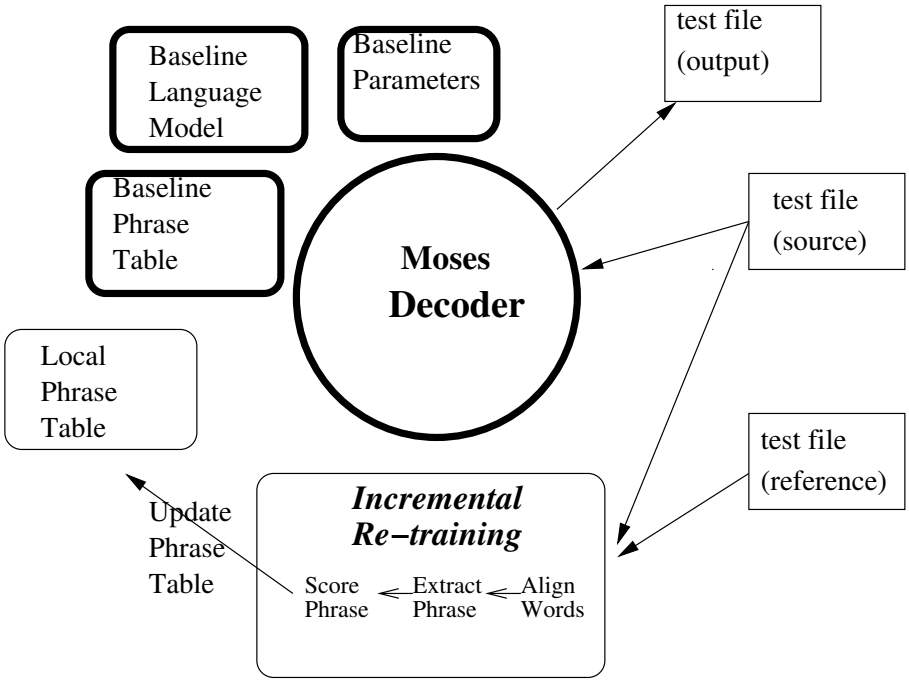

Figure 1: Simulated Postediting – decode source sentence, then update local phrase table with source sentence together with reference translation.

It is worth noting that the files in the protocol and patent data have a fairly well-defined structure – a single file will deal with one overall topic, with a good deal of specific terminology associated with the topic. This is not the case with the Europarl data. Here, each file is simply one day of proceedings. Each day can include multiple topics of discussion, and a topic can also continue from one day to the next.

For each translation domain, the baseline system consists of the all the translation data except for a set of files reserved for the purposes of testing and incremental re-training.

3.2 Training

The training and translation were performed with the standard Moses software (Koehn et al., 2007). This includes the standard optimization step in training, which we performed only with the *Baseline1* system. The other systems all use the Baseline1 parameters. The additional phrase table used by the other systems used a copy of the phrase table parameters obtained in the Baseline1 optimization. This gives an advantage to the Baseline1 system. Still, we chose this in the interests of simplicity, since the dynamic nature of the incremental re-training process would introduce a complication into the standard optimization process.

We translate each test file with four different systems:

1. **Baseline1:** a system not including translation data from any testfiles

2. **Baseline2:** here, the system includes a local phrase table built from a single testfile, but not the same file as the one being translated.

3. **Approximate:** each translated sentence is used to extend the local phrase table. Here, the alignments are produced by our quick, approximate technique

4. **Oracle:** each translated sentence is used to extend the local phrase table, using oracle alignments (produced by a complete GIZA++ training)

The results below show that incremental retraining can indeed take advantage of incrementally produced training data.

Table 1: Protocols – Bleu Scores

File	Base line1	Base line2	Approx-imate	Oracle
0	88.15	86.80	88.04	88.19
1	48.53	48.21	50.24	52.45
2	66.00	67.82	68.71	71.53
3	57.23	57.25	58.27	59.33
4	71.66	71.62	73.09	76.09
5	56.72	54.61	57.97	64.45
6	66.75	66.52	66.87	73.60
7	75.80	77.25	77.10	77.38
Avg	66.36	66.25	67.54	**70.38**
Avg BLEU change			2.84	**4.02**

Table 2: Patents – Bleu Scores

File	Base line1	Base line2	Approx-imate	Oracle
0	61.61	61.43	62.06	65.42
1	62.39	62.04	64.01	67.63
2	64.55	64.11	65.49	69.67
3	53.51	53.55	54.21	58.56
4	74.92	74.42	75.65	77.67
5	59.51	59.77	60.68	65.66
6	79.07	79.29	79.44	81.75
7	65.87	66.01	66.67	69.60
Avg	65.17	65.08	66.03	**69.50**
Avg BLEU Change			0.86	**4.33**

3.3 Protocols

Table 1 gives BLEU score results for each of the 8 testfiles in the protocol domain.

The system with the oracle alignments has a 4.02 BLEU increase over the Baseline1 system. The Oracle result shows that there is a strong file-context effect in this data. In other words, the translation data within a given file has much greater value for translation of the rest of the file than translation data from other files. This is clearly demonstrated by examining the Baseline2 system, which includes translation data from one of the test files in addition to the Baseline1 data. Thus on average the Baseline2 systems have as much data as the incremental trained systems – the difference, we claim, must be attributed to the file-context effect. Furthermore, we show that the Approximate Alignment method also gives a substantial improvement of 2.84 over Baseline1. This shows that the method is able to effectively exploit the file-context effect, but it also shows that improved online alignment techniques would result in substantial additional improvements.

3.4 Patents

Here, we present results for Patent data. Again, we have translated each of the 8 testfiles with online extension of the phrase table, using both oracle and approximate alignments, and we have defined Baseline1 and Baseline2 as above.

Table 3: Europarl – Bleu Scores

File	Base line1	Base line2	Approx-imate	Oracle
0	31.63	31.48	31.69	31.92
1	31.88	31.91	31.79	32.01
2	35.00	34.93	35.02	35.35
3	35.29	35.25	35.29	35.54
4	33.49	33.24	33.56	33.82
5	27.52	27.52	27.58	27.72
6	27.89	27.81	27.98	28.13
Avg	31.81	31.73	31.84	**32.07**
Avg BLEU Change			0.03	**0.26**

Again we see a strong confirmation of the file-context effect – the oracle alignment system has a 4.33 BLEU increase over the Baseline1 system. Again, the system with approximate alignments has a smaller improvement of 0.86.

3.5 Europarl

Here, we present results for Europarl data(Koehn, 2005). Again, we have translated each of the testfiles with online extension of the phrase table, using both oracle and approximate alignments, and the Baseline1 and Baseline2 systems.

Here there is an improvement of 0.26 over Base-

line1 with Oracle Alignments – this is a much weaker effect than that observed with patents and protocols. One possible explanation of this has to do with the structure of the Europarl files: as mentioned above, the technical patent and clinical trial protocol files have well defined structures and are clearly organized around a topic or group of topics. Europarl files do not exhibit such a structure – a single file can easily contain a variety of discussions of different topics, and conversely a topic may extend over more than one file.

4 Sentence Match and File-Context Effect

Above, we have presented evidence that there is a strong file-context effect with translation data. That is, translation data produced in a file has great value in translating subsequent portions of that same file. This is similar to an effect well-known among professional translators, concerning the use of *translation memory* systems. Such systems store previously translated sentences in a sentence database called a *translation memory*: this can be used in an interactive translation environment so that the translator retrieves the previous translation for any repeated sentences. Translation memory systems provide a substantial boost to productivity with texts involving many repeated sentences, and many translators build up quite large translation memories over time. However, even when using a large translation memory, there are often disproportionately many matches within the file.

Our own observation of this effect was in part what inspired us to pursue the work described in this paper. However, it also raises the question: to what extent is the file-context effect described here merely a reflection of the frequency of repeated sentences? If it were largely a reflection of repeated sentences, the effect would be no less real, but there would be no need to modify an SMT system to take advantage of it: instead, it would merely be evidence for the utility of TM systems, perhaps in concert with an SMT system.

In Tables 4 - 6 we present data showing that the file-context remains substantial even when repeated sentences are removed. We report on translations produced by the same four systems as above. The only difference is that any repeated sentences are removed.

Table 4: Protocols without Matching Sentences – Bleu Scores

File	Base line1	Base line2	Approx-imate	Oracle
0	88.09	86.76	88.00	87.75
1	47.96	47.63	49.57	51.99
2	66.08	67.84	67.98	69.91
3	56.69	56.73	57.40	58.42
4	71.71	71.64	72.08	74.09
5	56.90	54.56	57.62	62.70
6	67.77	67.50	67.59	72.30
7	75.06	75.01	75.90	76.02
Avg	66.28	65.96	67.02	69.15
Avg BLEU change			0.74	**2.87**

Overall, the average BLEU score change is nearly identical with both Patents and with Europarl. It is noticably smaller with Protocols – with oracle alignments the BLEU change is down to 2.87 from 4.02. But this is still a substantial improvement.

5 Recency

The overall intuition behind the experiments discussed above is that translation data from the same document has a much higher value to SMT of a sentence than other translation data from the same domain. In this section, we take this one step further by examining whether translation data of the immediately preceeding context of the same document has more value to SMT than data from earlier in the document.

The experiments are conducted by first splitting each test document into 20 parts of roughly the same word count. The average size of a part over the source documents is 1,585 words for protocols, 721 for patents, and 3,587 for Europarl. For each of these parts, an SMT system is created which is an extension of the baseline system with the additional phrases learned in this part. The phrase extraction is based on the oracle alignment in this experiment.

The last 10 parts of each document are then translated with systems trained with data from the preceeding 10 parts one at a time. For example, part 15

Table 5: Patents without Matching Sentences – Bleu Scores

File	Base line1	Base line2	Approx-imate	Oracle
0	60.94	60.79	61.26	64.51
1	62.03	61.67	63.37	67.09
2	62.18	61.87	63.09	67.46
3	53.30	53.34	54.18	58.36
4	73.96	73.46	74.71	76.71
5	60.01	60.19	61.07	65.87
6	78.62	78.88	79.04	81.62
7	65.17	65.34	66.40	68.91
Avg	64.53	64.44	65.39	68.82
Avg BLEU change			0.86	**4.29**

Table 6: Europarl without Matching Sentences – Bleu Scores

File	Base line1	Base line2	Approx-imate	Oracle
0	31.56	31.41	31.63	31.85
1	31.80	31.82	31.71	31.92
2	34.99	34.92	35.00	35.33
3	35.21	35.17	35.20	35.45
4	33.42	33.17	33.49	33.75
5	27.45	27.48	27.54	27.69
6	27.94	27.75	27.93	28.09
Avg	31.77	31.67	31.79	32.01
Avg BLEU change			0.02	**0.24**

is translated with each of the systems for parts 5–14, where 14 will represent the most recent context and 5 the most distant. This provides 10 levels of *contextual distance* for each part, where a system with low contextual distance will have phrase information from text close to the part being translated.

This results in 100 measuring points for each document, 10 for each contextual distance. Since we use the 8 test documents, we end up with 80 measuring points for each contextual distance for each domain.

Figure 2 shows the average increase in BLEU% over the baseline system when adding phrases from a part at a given contextual distance to the part being translated. This shows that adding relatively small amounts of data compared to the entire amount of training data, we see a substantial increase in BLEU% when the information comes from the immediate context. Just as with the file context effect, we see a very strong effect for protocols and patents, and a very small one for Europarl.

These results clearly support the hypothesis that more recent translation data has a higher value to the translation of a sentence within the same document than more distant translation data.

We expect to utilize the effect of recency more directly in future experiments by adding a recency parameter to the translation system. This could done by penalizing phrases according to the distance in sentences to the origin of their extraction. By adding an upper limit to the penalty, that is lower than the penalty added to phrases from the baseline phrase table, all within-document extracted phrases are promoted, but recent phrases are promoted the most.

6 Novel Repeat n-gram Percentage

We have seen that the value of translation data can vary widely, and we wish to understand the cause of this variation. Since we are using an SMT system which is based on n-grams, it is natural to expect that the value of translation data has to do with n-gram occurrences. In general, one might expect an SMT system to perform better with previously seen n-grams than with new n-grams. Thus we would expect that translation data would have value to the extent it converts new n-grams to previously seen ones. Following this reasoning, we propose to count

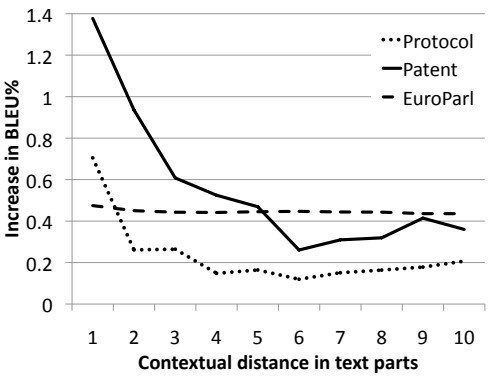

Figure 2: Recency effect experiments showing the average increase in BLEU% over the baseline system when adding phrases from the part at a given contextual distance. A contextual distance of 1 reflects the immediately preceeding text part, and higher values are further back.

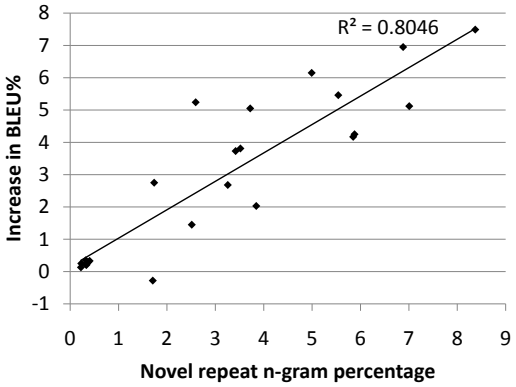

Figure 3: Novel repeat n-gram percentage compared to increase in BLEU% for all test documents from all domains used in the experiments. Correlation significant with $p < 0.0001$.

occurrences of *novel repeat n-grams*. These are n-grams that are *novel* in the sense that they do not appear in the baseline phrase table, and they are *repeats* because they occur at least twice in the file to be translated. This means that, with incremental re-training, the repeat occurrences will no longer be new events for the decoder.

We will study the correlation between increase in BLEU and the *novel repeat n-gram percentage* of the document. The novel repeat n-gram percentage is calculated by first counting how many of the n-grams in a document appear in an earlier sentence but not in the phrase table. This number is then divided by the total number of n-grams in the document.

If S is the set of n-grams occurring in a given line, PT is the set of source n-grams in the baseline phrase table, and A is the set of source n-grams occurring in earlier sentences, the novel repeat n-grams in s defined by $S_{nrn} = S \cap (A - PT)$.

If n is the n-gram length with an upper limit N, $g(n)$ is an n-gram of length n, s is the present sentence, m is the number of sentences, and S_{nrn} are the novel repeat n-grams in s, then the novel repeat n-gram percentage is

$$NRN\% = \frac{100}{N} \sum_{s=1}^{m} \sum_{n=1}^{N} \frac{\sum_{g(n) \in S_{nrn}} count(g(n))}{\sum_{g(n) \in S} count(g(n))}$$

In short, the metric describes the amount of n-gram coverage that is gained by extending the original phrase table as the document is post-edited.

In this analysis, we joined the data from all three domains to see if there was a correlation between the increase in n-gram coverage and the increase in BLEU. This provides us with 24 measuring points (3 times 8 test documents). For each test document, we extracted the novel repeat n-gram percentage and the increase in BLEU. Figure 3 shows the linear regression for this combined data set together with the coefficient of determination, R^2, which revelas a strong and highly significant correlation ($R^2 = 0.8046$, $p < 0.0001$). We thus expect that novel repeat n-gram percentage can be used as metric to predict the impact on translation quality for a given document, when extending a phrase table with a given data set.

This result suggests that it might be possible, not only to predict the impact of translation data in the context of incremental re-training, but more generally for evaluating whether adding a given data set to an existing system will benefit the translation of a given document.

7 Practical Implementation of Incremental Re-training

The incremental re-training prototype currently takes several seconds per sentence for the modest sized protocol system, and perhaps a minute per sentence for the larger patent and Europarl systems. A practical implementation will require substantial

further improvements. We believe that the necessary improvements can be implemented in a straightforward manner. In the current implementation, all required phrases are extracted and rescored each time an extended phrase table is constructed to accomodate a new sentence. However, only a very small number of phrases need to be examined – the vast majority of phrases can simply be left unchanged. This would vastly speed up the incremental re-training process. We plan to tightly integrate incremental re-training with the decoder. By doing this, and providing an efficient representation of the phrase options, we expect to achieve sufficiently fast response time for an interactive setting.

The approximate alignments can also be computed more quickly and more accurately. Here, we may be able to take advantage of recent work in which GIZA++ is re-conceived as an incremental process. This work is briefly examined in the next section.

8 Related Work

(Levenberg et al., 2010) describes an online version of the EM algorithm for word alignment. This version, they claim, "achieves the same performance" as traditional batch re-training. This approach is applied in a scenario with large amounts of continuous incoming data. It is not clear whether it could be adapted to the interactive setting contemplated in the present paper. If it could be so adapted, this might make it possible to achieve the same value as the oracle alignments in the online setting.

(Levenberg et al., 2010) also present results concerning recency effects which are somewhat similar to the effects described in the present paper, although these effects are not with respect to data within a given file, but over a period of days of Europarl proceedings. They point out that such a recency effect "underlines the need to retrian translation models with timely material". The recency effects in the present paper provide further support for this view.

9 Conclusions and Future Work

It is generally acknowledged that domain-specific training data is important for the translation quality of an SMT system (Bertoldi and Federico, 2009). This is presumably because in-domain training data resembles the text to be translated more than out of domain data. This resemblance both reduces the likelihood of unseen events in the input text, and increases the likelihood of correct translation choices for ambigous words and phrases.

In this paper, we have shown that there can be a great deal of difference in the value of different sets of domain-specific translation data. In particular, we have demonstrated two effects: *file-context*, which means that data within the same file has a much greater value, and *recency*, according to which more recent data within the file has greater value. Furthermore we have presented evidence that these differences in the value of translation data are strongly correlated with what we call *novel repeat n-grams*: n-grams not in the baseline data that are repeated in the new data. We believe this points to the possibility of automatically assessing the potential value of translation data for a given test corpus.

These results provide the basis for several interesting research avenues. First, we will continue to explore the file-context and recency effects we have found. Second, we will investigate the *novel repeat n-gram percent* to see whether it can be used to consistently predict the value of translation data. Finally, we will work towards the practical exploitation of this work by integrating incremental retraining into an interactive post-editing system. Below we make some remarks on these three directions for future work.

Concerning file-context and recency effects, there are several ways in which these effects might be made to emerge more clearly. One obvious limitation of the approach described here is that we incrementally update the phrase table, but we leave the language model unchanged. It seems reasonable to expect that incrementally updating the language model would strengthen the observed effects. Furthermore, we have not optimized parameters for the incrementally re-trained systems. We suspect that the system would benefit from information about which data comes from the same file, and where it occurs. If data was labeled in this way, it would be possible to optimize the system to provide perhaps a higher weight to phrase table entries based on such criteria.

Such modifications may well lead to stronger file-

context and recency effects. We also intend to examine the structure of the training data more closely. Recall that these effects were very weak with the Europarl data. One possible explanation for this is that files do not correspond to topical units in the way they typically do in technical domains. Instead, each file represents a single day of proceedings. A single file will typically contain discussions of different topics, and a single discussion might also continue from one file to the next. We intend to experiment with different ways of dividing the Europarl data into "files": one possibility is to use the html markup supplied with the Europarl files to define topical sections. The *Chapter ID* labels would appear to be appropriate for this purpose. Another possibility would be to use statistics on word or n-gram occurrences to define sections, perhaps according to an algorithm such as textTiling (Hearst, 1997).

The second direction concerns *novel repeat n-gram percent*, a metric we have found is strongly correlated with translation value. The intuition behind this metric is the simple idea that, other things being equal, an SMT system does better with old events than with new events. We found a strong correlation between this metric and the value of translation data. We would like to see if this metric could be used to predict the value of some translation data. This might be used to automatically select data for a domain-specific corpus. Or data might be weighted more highly if it receives a high score on this metric.

The third direction for future work concerns the practical development of a system in which incremental re-training is integrated in an interactive post-editing system. One important aspect of this work involves the approximate alignments currently used in incremental re-training. These alignments are of poorer quality than those produced by a traditional GIZA++ re-training, although, as we have seen, they are of a quality that is sufficient to provide substantial benefit to translation quality. The simple heuristics we employ can undoubtedly be improved – one promising approach to this is based on the work of (Levenberg et al., 2010), who suggest that their approach to on-line word alignment can match the quality of the standard GIZA++ approach. We plan to investigate whether this work could be applied to our incremental re-training scenario.

The incremental re-training system we have developed has made it possible to perform the experiments reported in this paper: without incremental retraining, it would be difficult or impossible in practice to perform large numbers of experiments on the effect of different translation data in hundreds of different situations. We are now planning to develop the system further, so that incremental re-training could be deployed in real time as part of an interactive post-editing system. By building our prototype incremental retraining system, we have shown that a real-time incremental re-training system is possible. Furthermore, we have shown that recent, file-internal translation data can have great value. This suggests that a practical incremental re-training system would have great impact on the quality of SMT systems.

Acknowledgments

Thanks to Novartis Denmark for providing the Clinical Protocol translation data, and to Lingtech A/S for providing the Patent translation data.

References

N. Bertoldi and M. Federico. 2009. Domain adaptation in statistical machine translation with monolingual resources. In *Proceedings of EACL, WMT*.

W. Gale, K. Church, and D. Yarowsky. 1992. One sense per discourse. In *Proceedings of the 4th DARPA Speech and Natural Language Workshop*, pages 233–237.

Marti A. Hearst. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL, demonstation session.*

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit*.

Abby Levenberg, Chris Callison-Burch, and Miles Osborne. 2010. Stream-based translation models for statistical machine translation. In *Proceedings of NAACL.*

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):607–615.