CombAlign: a Tool for Obtaining High-Quality Word Alignments

Steinþór Steingrímsson
Department of
Computer Science
Reykjavik University
Iceland
steinthor18@ru.is

Hrafn Loftsson
Department of
Computer Science
Reykjavik University
Iceland
hrafn@ru.is

Andy Way
ADAPT Centre
School of Computing
Dublin City University
Ireland
andy.way
@adaptcentre.ie

Abstract

Being able to generate accurate word alignments is useful for a variety of tasks. While statistical word aligners can work well, especially when parallel training data are plentiful, multilingual embedding models have recently been shown to give good results in unsupervised scenarios. We evaluate an ensemble method for word alignment on four language pairs and demonstrate that by combining multiple tools, taking advantage of their different approaches, substantial gains can be made. This holds for settings ranging from very low-resource to high-resource. Furthermore, we introduce a new gold alignment test set for Icelandic and a new easy-to-use tool for creating manual word alignments.

1 Introduction

Word alignment, the task of finding corresponding words in a bilingual sentence pair (see Figure 1), was a key component of statistical machine translation (SMT) systems. While word alignments are not necessary for neural machine translation (NMT), various MT methods incorporating word alignment have been found to achieve significant improvements in performance. Alkhouli et al. (2018) and Liu et al. (2016) use alignments as a prior; Arthur et al. (2016) augment NMT systems with discrete translation lexicons that encode low-frequency words; Press and Smith (2018) infer a correspondence between words in sentence pairs before encoding/decoding and, as demonstrated by Poncelas et al. (2019), back-translated data created using SMT systems, requiring word alignments, can be valuable to augment NMT systems. Word alignments have also been utilized to improve automatic post-editing (Pal et al., 2017) as well as to preserve markup in machine-translated texts (Müller, 2017).

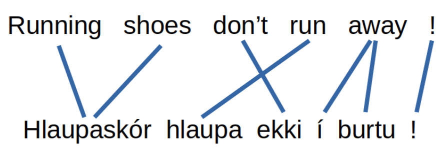

Figure 1: A simple example of English-Icelandic word alignments. Corresponding words are connected by edges.

Various other subfields of NLP make use of word alignments. Shi et al. (2021) show that by simply pipelining word alignment with unsupervised bitext mining, bilingual lexicon induction (BLI) quality can be improved significantly. For BLI, Artetxe et al. (2019) use an unsupervised MT pipeline, also employing word alignments. Kurfalı and Östling (2019) use word alignments to filter noisy parallel corpora, and Paetzold et al. (2017) include word alignment as a part of their pipeline to align monolingual comparable documents.

There is a variety of word aligners available. *Giza++* (Och and Ney, 2003) and *fast_align* (Dyer et al., 2013) are easy to use implementations of the IBM models (Brown et al., 1993). Other statistical aligners, such as *eflomal* (Östling and Tiedemann, 2016), have also been shown to be fast and give competitive results. *SimAlign* (Masoud et al., 2020) takes advantage of the rising availability of contextualized embeddings and leverages them by extracting alignments from similarity matrices.

In this work, we present *CombAlign*, an ensemble of these four tools (Giza++, fast_align, eflomal, and SimAlign). As they are based on different approaches, and all able to attain a fairly high F_1-score, it is reasonable to expect that combining their results in a sensible way could give better results than using any one of the individual systems.

Recently, the first reported results in SMT and NMT for Icelandic were published (Jónsson et al., 2020) within the context of an Icelandic national language technology programme (Nikulásdóttir et al., 2020). Icelandic is a morphologically rich West Germanic language with relatively few speakers, for which a substantial amount of language resources has been made available in recent years. However, no previous work has been conducted on word alignments for Icelandic. While testing our methods on four language pairs, we focus in particular on the effects of different alignment methods on the English-Icelandic (en-is) language pair. For finding the best hyperparameters for our ensemble, we thus do a grid search using an en-is development set.

Our main contribution is showing that it is possible to obtain high-quality word alignments using a combination of selected tools, outperforming all of the individual word alignment tools. We show this for four language pairs, with more detailed scrutiny of the results for one of them, en-is. Furthermore, we:

- publish a new gold standard word alignment reference set for en-is.
- make available a graphical tool, *AlignMan*, for manually curating word alignments.[1]
- make the source code available for running the alignment tools and extracting combined alignments from them.[2]

2 Related Work

The most common statistical word alignment tools are based on the IBM models (Brown et al., 1993). These include fast_align (Dyer et al., 2013), Giza++ (Och and Ney, 2003) and eflomal (Östling and Tiedemann, 2016), all used in this work. The five IBM models use lexical translation probabilities and probability distributions with the different models adding or emphasizing different features to tackle weaknesses of the other models. While fast_align builds on IBM model 2, Giza++ iterates on a number of the models in sequence, as well as using an HMM model. eflomal uses a Bayesian model with Markov Chain Monte Carlo inference on the IBM models.

Several studies on word alignments in relation to neural models have been published. Liu et al. (2016) show that attention can be seen as a reordering model as well as an alignment model, and Ghader and Monz (2017) investigate the differences between attention and alignment. Zenkel et al. (2019) apply stochastic gradient descent to directly optimize the attention activations towards a given target word, resulting in accurate word alignments, and Garg et al. (2019) extract discrete alignments from the attention probabilities learnt during regular NMT training and leverage them to optimize towards translation and alignment objectives. Most of these systems require parallel data for training, but SimAlign (Masoud et al., 2020) takes advantage of the rising availability of contextualized embeddings and leverages them by extracting alignments from similarity matrices induced from the embeddings, with no need for any external data.

Ensemble methods are common in NLP and, in many cases, have been shown to give more accurate results than using just one single approach. They have been used, for example, for classifying patent applications (Benites et al., 2018), for spellchecking (Stefanescu et al., 2011), POS-tagging (Henrich et al., 2009) and sentiment analysis (Araque et al., 2017). For word alignments, Tufiş et al. (2006) have previously used a union of two different alignment approaches, each producing distinct alignments. One of their aligners was an implementation of the IBM models, and the other used translation lexicons and phrase boundaries to detect alignments. Their combined aligner outperformed both individual systems, and its results produced approximately 10% fewer errors than the better individual aligner.

3 Data

For evaluation, we use gold standard word alignments for four language pairs: Czech, German, French and Icelandic, all paired with English (en-cs, en-de, en-fr and en-is, respectively). For the methods trained on parallel data, Giza++, fast_align and eflomal, we use a subset of 512k sentences from Europarl (Koehn, 2005), except in the case of Icelandic as detailed in Section 3.1. Further information on the test sets is given in Table 1.

3.1 Icelandic Data

No gold standard word alignments have previously been made available for Icelandic. In order

[1] `https://github.com/steinst/AlignMan`
[2] `https://github.com/steinst/CombAlign`

Lang. Pair	Gold Standard	Sent. Pairs	Edges
en-cs	Mareček (2008)	2,501	67,424
en-de	Europarl[3]	508	10,534
en-fr	Och and Ney (2000)	447	17,438
en-is	*new*	384	5,517

Table 1: Gold standard alignments used for evaluation. The en-is gold standard contains further 220 sentence pairs that were used as a development set for grid search.

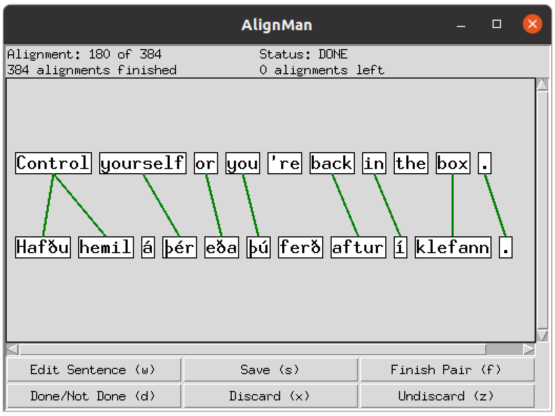

Figure 2: A screenshot from AlignMan. The program reads in text files with parallel sentences. The user can edit the sentences, discard them or create edges between words by moving the cursor to select corresponding words and then saving the alignment. It supports up to two users and can export a union or intersection of their alignments in two different formats.

to test our approach and other alignment methods on Icelandic, we thus compiled development and test sets. For that purpose, we created a simple graphical tool for performing manual word alignment, *AlignMan*, which is available under an Apache2 licence. A screen shot from AlignMan can be seen in Figure 2.

Two annotators manually aligned 604 sentences, a random sample from the *ParIce* en-is parallel corpus (Barkarson and Steingrímsson, 2019). They then reviewed the other annotator's work in order to eliminate mistakes. The two annotations were then combined. All 1-to-1 alignments that the annotators agreed upon were marked as 'sure' alignments and all other alignments made by either one or both of the annotators were marked as 'possible' alignments. The set was then split in two, with 220 sentences in a *dev*-set and 384 sentences in a *test*-set. The gold alignments are available for download from the CLARIN repository[4] where further information on the criteria for building the corpus is available.

When parallel data was required to train the word aligners, sentence pairs from the ParIce corpus were used.

4 Methodology

In order to find the best settings for each aligner, we carry out a grid search. We run Giza++, fast_align and eflomal using different setups. For SimAlign, we use two different contextual embedding models and run them with different hyperparameters. We are thus working with five different aligners/alignment models. Finally, we proceed to find the best ensemble for different levels of parallel data availability.

4.1 Experimental Setup

By default, Giza++ runs IBM models 1, 3 and 4 as well as an HMM model, while fast_align is based on IBM model 2. We use default settings for these two aligners as well as for eflomal and compared their results after processing their output with different heuristics. These aligners are not trained on other word alignments, but rather on sentence-aligned parallel texts. They use an expectation maximization algorithm, iteratively learning from the parallel sentences; starting by initializing the model, then applying it to the data and setting the most probable alignments. After filling in gaps and collecting counts for particular word translations a new probability distribution is estimated. These steps are iterated until convergence.

Because the aligners learn probabilities from the data they run on, they should be better able to induce lexical translation probabilities and probability distributions when the size of the data increases, which in turn should lead to an increase in quality. In order to study this effect, we ran the aligners with varying numbers of sentences. The data we use for the experiments is described in Section 3.

[3] `https://www-i6.informatik.rwth-aachen.de/goldAlignment/`

[4] `http://hdl.handle.net/20.500.12537/103`

Giza++	
All settings default	
fast_align	
Heuristics	**intersection**, union, gd, gdf, gdfa
eflomal	
Heuristics	**intersection**, union, gd, gdf, gdfa
SimAlign	
Models Tokenization Heuristics Distortion Null extension	BERT, **XLM-R** **Word**, BPE **Argmax**, Itermax, Match [0.02, 0.03, ..., **0.09**, ..., 0.15] [0.85, 0.90, 0.95, 0.96, 0.97, 0.98, 0.99, **1.0**]

Table 2: Hyperparameters for the different aligners. Shown in bold are the ones giving the highest F_1-score.

Giza++ only outputs one set of alignments after each run, but for fast_align and eflomal we output alignments for both directions, source→target language and target→source, and then generate alignments from these using different alignment heuristics: intersection and union, as well as grow-diag (gd), grow-diag-final (gdf) and grow-diag-final-and (gdfa).

With SimAlign, we induce alignments from two different contextualized embedding models, multilingual BERT (mBert) (Devlin et al., 2019), and XLM-R (Conneau et al., 2020), and run experiments both for whole words and byte-pair encodings (BPE) (Sennrich et al., 2016). The alignments are obtained from similarity matrices using three different methods: *Match*, a graph-based method that identifies matches in a bipartite graph; *Argmax*, which aligns two words if the target word is the most similar to the source word, or vice versa; and *Itermax*, which applies Argmax iteratively and is thus better able to find alignment edges when one word aligns with two or more words in the other language. We did a grid search on the en-is development set, calculating the best scores using these methods and two other hyperparameters: distortion correction and null extensions, which set a threshold for when to remove edges and create null alignments. Different settings in our grid search are shown in Table 2.

For each of the alignment tools, we selected the hyperparameters giving the highest F_1-score. Then another grid search was carried out to find how best to combine the results. For that we had two parameters: combination of alignment tools, with 3 to 5 aligners/alignment models in each ensemble; and different parameters to join the alignments: with `unionall`, which accepts all alignments of the systems in the suggested ensemble, and different levels of intersection, from `intersectmin2` that requires two aligners to agree for an edge to be accepted, to `intersectmin5` where all aligners have to agree on each edge.

Finally, in order to examine whether our ensemble method is applicable to other language pairs, we test it on three of the test sets used in Masoud et al. (2020) and compare our results to theirs.

5 Experiments and Results

As described in Section 4.1, we identified the optimal settings and post-processing heuristics for each tool using grid search on the *dev*-set (see Table 2). We used these settings to obtain scores on our *test*-set, as shown in Tables 3 and 4.

5.1 Individual Aligners

While we use the same setting for each tool throughout, after having executed the grid search, the results of the ensemble differs in relation to how much data is being aligned. Relying at least in part on lexical translation probabilities, fast_align and Giza++ require a substantial amount of data before they become fairly accurate, while eflomal seems to be less susceptible to paucity of data. Figure 3 shows how F_1 increases for each system when evaluated on the Icelandic test set, when more parallel sentences are added for training. The aligners always learn from at least 384 test sentences, and up to an additional 3.6 million sentences. Table 3 shows precision, recall, F_1-score and number of edges, i.e. individual word alignments, produced by eflomal, Giza++, and fast_align, when run with varying numbers of sentence pairs. Rather accurate from the start, the main advantage of training eflomal on more data is to get higher recall and more edges, while Giza++ and fast_align always output a similar number of edges, but both precision and recall rise when more sentence pairs are added.

SimAlign does not need any parallel data to learn from, and unlike the other aligners the results do not change when there is more data to

	eflomal intersect				Giza++				fast_align intersect			
Samples	Prec.	Rec.	F_1	Edges	Prec.	Rec.	F_1	Edges	Prec.	Rec.	F_1	Edges
0	.85	.76	.80	3803	.62	.74	.67	5387	.73	.67	.70	4005
1000	.87	.81	.84	4003	.64	.74	.68	5247	.78	.71	.74	3979
2000	.87	.83	.85	4098	.64	.75	.69	5223	.80	.73	.76	3978
4000	.87	.85	.86	4229	.64	.74	.68	5143	.82	.75	.78	3978
8000	.87	.87	.87	4320	.65	.74	.69	5117	.83	.76	.80	3976
16000	.88	.89	.88	4432	.67	.77	.72	5089	.85	.78	.81	3998
32000	.88	.90	.89	4507	.70	.79	.74	5072	.87	.80	.83	4008
64000	.88	.92	.9	4561	.72	.82	.77	5051	.88	.82	.85	4034
128000	.88	.93	.91	4622	.75	.85	.80	5019	.89	.84	.87	4086
256000	.88	.93	.91	4654	.78	.87	.82	5000	.90	.85	.88	4139
512000	.88	.93	.91	4667	.81	.89	.85	4982	.90	.86	.88	4151
1024000	.88	.94	.91	4713	.83	.91	.86	4951	.91	.87	.89	4165
2048000	.88	.94	.90	4722	.84	.91	.87	4927	.91	.86	.89	4139
3600000	.88	.94	.91	4745	.85	.92	.88	4913	.91	.86	.89	4115

Table 3: Precision, recall, F_1-scores and number of edges for each of the IBM model-based aligners, with various numbers of parallel sentences added for training the aligners.

align. However, the tokenization used (BPE or the original word forms) and how the alignments are obtained from the similarity matrix, has a substantial effect on the resulting alignments, as seen in Table 4. The table shows that ArgMax gives a substantially higher precision than IterMax and Match, but since IterMax has higher recall, the F_1-scores are quite close.

5.2 Ensembles

As can be seen in Table 3, eflomal does not need much training data to reach high precision. Thus, it should not be surprising that in low-resource scenarios a combination of eflomal with the two unsupervised SimAlign models gives the best results. When more data is available, the other two IBM-model based aligners become more accurate, and as a consequence, more useful in an ensemble.

We thus report on two different ensembles: *EnsembleSmall*, comprised of three aligners which is better in cases where there is scarce data, and *EnsembleLarge* which uses all five aligners. Our ensemble strategy is simple: for both ensembles we only require a majority vote on each alignment. For EnsembleSmall we thus require 2 out of 3 aligners to suggest an alignment candidate for it to be accepted. EnsembleSmall uses the alignments produced by SimAlign's *IterMax*, which

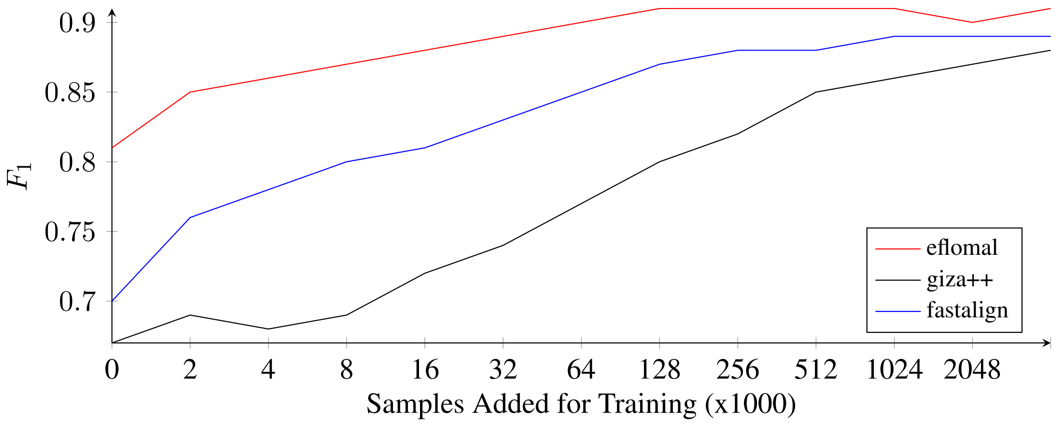

Figure 3: F_1 for word alignments generated using different alignment tools as a function of the number of sentence pairs used for training. F_1 for SimAlign-mBERT is 0.86 and 0.90 for SimAlign-XLM-R.

SimAlign						
Model	Tok.	H.	Pr.	Rc.	F_1	Edg.
mBERT	BPE	AM	.85	.84	.84	4468
		IM	.74	.91	.82	5717
		M	.66	**.92**	.77	6590
	word	AM	**.88**	.84	**.86**	4145
		IM	.79	.90	.84	5111
		M	.75	.91	.82	5463
XLM-R	BPE	AM	.88	.90	.89	4599
		IM	.78	.94	.86	5615
		M	.69	**.96**	.80	6618
	word	AM	**.92**	.88	**.90**	4165
		IM	.85	.93	.89	4925
		M	.78	.94	.86	5473

Table 4: Precision, F_1-measure and number of edges for different setups of SimAlign. All these settings use 0.09 for distortion. The heuristics are: AM=ArgMax, IM=IterMax, M=Match.

CombAlign					
Samples	Ensemble	Prec.	Rec.	F_1	Edges
0	EnsSm	.92	.92	.92	4410
	EnsLa	.93	.81	.87	3743
1000	EnsSm	.92	.93	.92	4458
	EnsLa	.94	.84	.89	3819
2000	EnsSm	.91	.93	.92	4459
	EnsLa	.95	.85	.90	3852
4000	EnsSm	.91	.93	.92	4506
	EnsLa	.95	.86	.90	3866
8000	EnsSm	.91	.94	.92	4529
	EnsLa	.95	.87	.91	3933
16000	EnsSm	.91	.94	.93	4569
	EnsLa	.96	.88	.92	3970
32000	EnsSm	.91	.95	.93	4591
	EnsLa	.96	.90	.93	4025
64000	EnsSm	.91	.95	.93	4624
	EnsLa	.96	.91	.93	4070
128000	EnsSm	.91	.95	.93	4635
	EnsLa	.96	.92	.94	4147
256000	EnsSm	.91	.95	.93	4656
	EnsLa	.96	.92	.94	4178
512000	EnsSm	.91	.95	.93	4648
	EnsLa	.96	.93	.94	4220
1024000	EnsSm	.91	.95	.93	4653
	EnsLa	.96	.94	.95	4249
2048000	EnsSm	.90	.95	.93	4679
	EnsLa	.96	.94	.95	4266
3600000	EnsSm	.90	**.95**	.93	4681
	EnsLa	**.96**	.94	**.95**	4265

Table 5: Precision, recall, F_1-scores and number of edges for different setups of the CombAlign ensemble.

has higher recall, an advantage when only one of the aligners in the ensemble is allowed to miss an alignment. EnsembleLarge requires 3 out of 5 aligners to agree and uses SimAlign's *ArgMax*, which has more precision. Figure 4 shows how the F_1-scores for the two ensembles rise with more data, and how EnsembleLarge, being more reliant on data, needs only tens of thousands of sentence pairs to outperform EnsembleSmall which obtains higher F_1-scores in very low-resource settings. In contrast, EnsembleLarge, always having higher precision as shown in Table 5, produces fewer edges.

Our combination is based on a majority vote and the ensemble obtaining the highest F_1-score is selected. Accordingly, it is possible to obtain higher precision using other combinations in situations where precision is critical and recall is not as important. This could be realised by setting a higher requirement for agreement between aligners, raising the precision even further, but at the price of retrieving fewer edges and thus a lower F_1-score. For higher recall, lowering the agreement requirements works, although at the cost of some precision. Table 5 shows the combinations giving the best precision and F_1-score, as well as recall and number of edges suggested by the system.

5.3 Utilizing the Word Alignments

As noted in Section 1, word alignments can be used for many different purposes, sometimes using SMT systems as intermediaries. In order to see whether our alignments are beneficial for SMT systems, we trained three Moses models, keeping all components of the training process the same, except for word alignments. For training, we used the data and filtering methods described in Jónsson et al. (2020).

Our baseline system uses the default Moses settings, with Giza++ for word alignments. We trained two other models, *CombAlignF1*: using the settings giving the highest F1-score as detailed in Section 5.2; and *CombAlignRec*: where we are still using the five aligners in the ensemble, but are more lenient and only require two or more of the five aligners to be in agreement. We did this as our highest scoring ensemble, *CombAlignF1*, generates 15% fewer edges than Giza++ and, for this task, recall is likely to be important. By relaxing the demands for agreement between the aligners, we raise recall while still only generating a similar number of edges between words as Giza++.

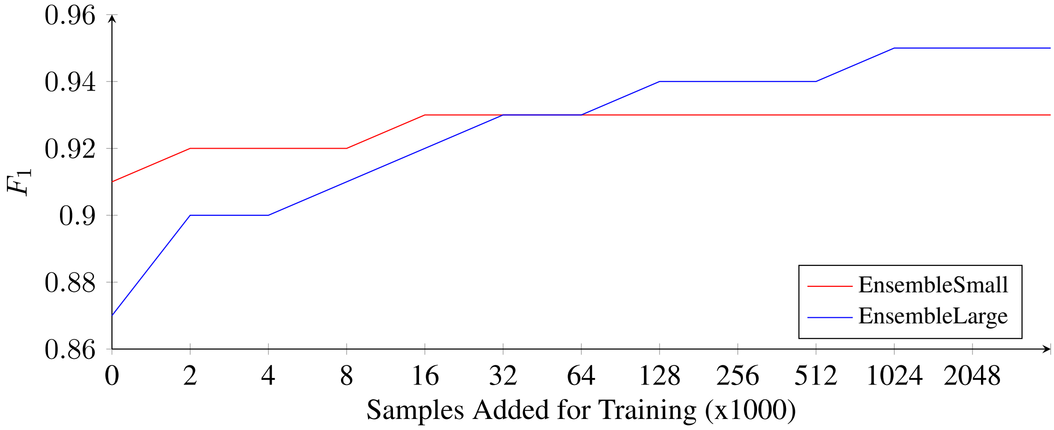

Figure 4: F_1 score for aligner ensembles. *EnsembleSmall* uses three alignment models and *EnsembleLarge* uses all five alignment models, as described in Section 5.2.

We compared these three systems in the following manner. First, we examined the phrase tables generated during training. The baseline system creates a phrase table with 3,496K lines, *CombAlignF1* has 1,319K lines and *CombAlignRec* has 1,774K lines. Manual inspection shows that the removed lines are almost always faulty so this pruning should not have negative effects on the system. Second, we tested the systems, using the three test sets from Jónsson et al. (2020), calculated the BLEU scores and manually inspected and evaluated the differences in translation.

BLEU scores for *CombAlignF1* were almost the same as for the baseline system, with a difference ranging from 0.01 to 0.11 for the three test sets. *CombAlignRec* had slightly better scores, scoring 0.4 to 0.95 higher BLEU than the baseline system.

We then manually compared a random sample of 450 translated sentences from the baseline system and *CombAlignRec*. 46% of the outputs were exactly the same; 14% had multiple faults for both systems and were deemed equally bad; 17% of the sentences were translated better by the baseline system and 23% had better translations produced by *CombAlignRec*. We categorized the errors made by the systems and while the sample size is quite small, and there is no clear distinction between the systems, *CombAlignRec* seems to be more likely to have errors when there are multiple numerical tokens in the sentence to translate, possibly because they may be treated like rare words. Moreover, *CombAlignRec* seems less likely to have words missing in the translated output and it seems more likely to make a more appropriate lexical choice, both in terms of content words and verb inflections. A more thorough investigation is needed to understand why this is the case.

5.4 Other Language Pairs

In order to show that the ensemble methods work for other languages than Icelandic, we ran an experiment on three test sets. Table 6 shows the results and a comparison to the previous best, as reported on in Masoud et al. (2020).

In this experiment, we used two settings for the IBM-model based alignment tools: only running on the test-set data, and running with additional parallel data of 512K sentence pairs for training each language pair. Although the results for CombAlign always outperform the individual aligners, the difference is not always as large as for the en-is language pair. This may possibly be explained by the fact that the contextualized embeddings have more data on the other languages and thus give better predictions than when predicting Icelandic, or that the parallel training data is not in the same domain as the test sets, while the Icelandic test sets contained sentence pairs sampled from the parallel corpus (ParIce) used for training.

For the best-scoring ensembles, we used SimAlign's *Itermax* when the statistical aligners used parallel data as well as when no additional data was used. This was due to Itermax giving the highest F_1-score for these language pairs. This was not true for Icelandic, possibly because the contextual models were trained on less Icelandic data and so have more 'knowledge' of these other languages than it has of Icelandic.

Method	cs-en		en-fr		en-de	
Train. data (K)	0	512	0	512	0	512
eflomal	.79	.86	.82	.91	.61	.73
fast_align	.66	.78	.73	.86	.52	.70
Giza++	.71	.81	.69	.89	.55	.73
SimAlign: XLM-R	.87		.93		.78	
SimAlign: BERT	.87		.94		.81	
Previous work	.87		.94		.81	
CombAlign	.89	**.91**	.95	**.95**	.80	**.83**

Table 6: Word alignment F_1-scores for cs-en, en-fr and en-de language pairs, with or without using training data.

6 Conclusion and future work

We have shown that using a very simple combination method for word alignment, it is possible to increase the accuracy substantially, both in low- and high-resource settings.

We evaluated on four language pairs, *en-cs*, *en-de*, *en-fr* and for the first time *en-is*, for which we manually created a new gold standard word alignment reference set. In order to do that we created and published a graphical tool for manual word alignments.

While our method uses minimal data processing, some pre-processing like POS-tagging and lemmatizing may raise the accuracy even further, especially in the case of a morphologically rich language like Icelandic. A comparison of typical misalignments per aligner is also likely to be beneficial, as knowing these properties might help in combining the aligners more effectively. The mBERT and XLM-R models we employ through SimAlign give good results, but there may still be room for improvement, for instance by pre-training these models on more Icelandic texts, which are scarce in the multilingual training corpus. It may also be worth considering to train a bilingual word embedding model and use that for alignment instead of, or in combination with, the other contextualized embedding models.

In the paper, we reported on preliminary results from training an SMT system using our word alignments. We plan to investigate whether the slightly better SMT output will be more beneficial for back-translations to augment NMT systems, following Poncelas et al. (2019). We also plan to compare BLI quality using the setup in (Artetxe et al., 2019) and the same setup using our alignments. Furthermore, we intend to apply our alignments to training alignment-assisted NMT transformer models, by adding an alignment attention layer as described in (Alkhouli et al., 2018).

Acknowledgements

This work is supported by the Language Technology Programme for Icelandic 2019-2023, funded by the Icelandic government, and by the ADAPT Centre for Digital Content Technology which is funded under the Science Foundation Ireland (SFI) Research Centres Programme (Grant No. 13/RC/2106) and is co-funded under the European Regional Development Fund.

References

Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney. 2018. On The Alignment Problem In Multi-Head Attention-Based Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 177–185, Brussels, Belgium.

Oscar Araque, Ignacio Corcuera-Platas, J. Fernando Sánchez-Rada, and Carlos A. Iglesias. 2017. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, 77:236 – 246.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019. Bilingual Lexicon Induction through Unsupervised Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5002–5007, Florence, Italy.

Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating Discrete Translation Lexicons into Neural Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567, Austin, Texas.

Starkaður Barkarson and Steinþór Steingrímsson. 2019. Compiling and Filtering ParIce: An English-Icelandic Parallel Corpus. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 140–145, Turku, Finland.

Fernando Benites, Shervin Malmasi, and Marcos Zampieri. 2018. Classifying Patent Applications with Ensemble Methods. In *Proceedings of the Australasian Language Technology Association Workshop 2018*, pages 89–92, Dunedin, New Zealand.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly Learning to Align and Translate with Transformer Models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China.

Hamidreza Ghader and Christof Monz. 2017. What does Attention in Neural Machine Translation Pay Attention to? In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 30–39, Taipei, Taiwan.

Verena Henrich, Timo Reuter, and Hrafn Loftsson. 2009. Combitagger: A system for developing combined taggers. In *Proceedings of the 22nd International FLAIRS Conference*, pages 254–259, Sanibel Island, Florida.

Haukur Páll Jónsson, Haukur Barri Símonarson, Vésteinn Snæbjarnarson, Steinþór Steingrímsson, and Hrafn Loftsson. 2020. Experimenting with Different Machine Translation Models in Medium-Resource Settings. In *Proceedings of Text, Speech, and Dialogue – 23rd International Conference*, volume 12284 of *Lecture Notes in Computer Science*, pages 95–103.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand.

Murathan Kurfalı and Robert Östling. 2019. Noisy Parallel Corpus Filtering through Projected Word Embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 277–281, Florence, Italy.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural Machine Translation with Supervised Attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan.

David Mareček. 2008. Automatic Alignment of Tectogrammatical Trees from Czech-English Parallel Corpus. Master's thesis, Charles University.

Jalili Sabet Masoud, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online.

Mathias Müller. 2017. Treatment of Markup in Statistical Machine Translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 36–46, Copenhagen, Denmark.

Anna Nikulásdóttir, Jón Guðnason, Anton Karl Ingason, Hrafn Loftsson, Eiríkur Rögnvaldsson, Einar Freyr Sigurðsson, and Steinþór Steingrímsson. 2020. Language Technology Programme for Icelandic 2019-2023. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3414–3422, Marseille, France.

Franz Josef Och and Hermann Ney. 2000. Improved Statistical Alignment Models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *Prague Bulletin of Mathematical Linguistics*, 106:125–146.

Gustavo Paetzold, Fernando Alva-Manchego, and Lucia Specia. 2017. MASSAlign: Alignment and Annotation of Comparable Documents. In *Proceedings of the IJCNLP 2017, System Demonstrations*, pages 1–4, Tapei, Taiwan.

Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, Qun Liu, and Josef van Genabith. 2017. Neural Automatic Post-Editing Using Prior Alignment and Reranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 349–355, Valencia, Spain.

Alberto Poncelas, Maja Popović, Dimitar Shterionov, Gideon Maillette de Buy Wenniger, and Andy Way. 2019. Combining PBSMT and NMT Backtranslated Data for Efficient NMT. In *Proceedings* *of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 922–931, Varna, Bulgaria.

Ofir Press and Noah A. Smith. 2018. You May Not Need Attention. *ArXiv*, abs/1810.13409.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Haoyue Shi, Luke Zettlemoyer, and Sida I. Wang. 2021. Bilingual Lexicon Induction via Unsupervised Bitext Construction and Word Alignment. *ArXiv*, abs/2101.00148.

Dan Stefanescu, Radu Ion, and Tiberiu Boros. 2011. TiradeAI: An Ensemble of Spellcheckers. In *Proceedings of the Spelling Alteration for Web Search Workshop*, pages 20–23, Bellevue, Washington.

Dan Tufiş, Radu Ion, Alexandru Ceauşu, and Dan Ştefănescu. 2006. Improved Lexical Alignment by Combining Multiple Reified Alignments. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 153–160, Trento, Italy.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2019. Adding Interpretable Attention to Neural Translation Models Improves Word Alignment. *ArXiv*, abs/1901.11359.