Low-Resource Unsupervised NMT: Diagnosing the Problem and Providing a Linguistically Motivated Solution

Lukas Edman, Antonio Toral, Gertjan van Noord
Center for Language and Cognition
University of Groningen
{j.l.edman, a.toral.ruiz, g.j.m.van.noord}@rug.nl

Abstract

Unsupervised Machine Translation has been advancing our ability to translate without parallel data, but state-of-the-art methods assume an abundance of monolingual data. This paper investigates the scenario where monolingual data is limited as well, finding that current unsupervised methods suffer in performance under this stricter setting. We find that the performance loss originates from the poor quality of the pretrained monolingual embeddings, and we propose using linguistic information in the embedding training scheme. To support this, we look at two linguistic features that may help improve alignment quality: dependency information and sub-word information. Using dependency-based embeddings results in a complementary word representation which offers a boost in performance of around 1.5 BLEU points compared to standard WORD2VEC when monolingual data is limited to 1 million sentences per language. We also find that the inclusion of sub-word information is crucial to improving the quality of the embeddings.

1 Introduction

Machine Translation (MT) is a rapidly advancing field of Natural Language Processing, where there is an ever-increasing number of claims of MT systems reaching human parity (Hassan et al., 2018; Barrault et al., 2019). However, most of the focus has been on MT systems under the assumption that there is a large amount of parallel data available, which is only the case for a select number of language pairs.

Recently, there have been approaches that do away with this assumption, requiring only monolingual data, with the first methods based solely around neural MT (NMT), using aligned pretrained embeddings to bootstrap the translation process, and refining the translation with a neural model via denoising and back-translation (Artetxe et al., 2017b; Lample et al., 2017). More recently, statistical MT (SMT) approaches as well as hybrid approaches, combining SMT and NMT, have proven more successful (Lample et al., 2018; Artetxe et al., 2019).

While the unsupervised approaches so far have done away with the assumption of parallel data, they still assume an abundance of monolingual data for the two languages, typically assuming at least 10 million sentences per language. This amount of data is not available for every language, notably languages without much of a digital presence. For example, Fulah is a language spoken in West and Central Africa by over 20 million people, however there is a scarce amount of data freely available online. This motivates a new paradigm in unsupervised MT: Low-Resource Unsupervised MT (LRUMT).

In this paper, we investigate the reasons why current unsupervised NMT methods fail in the low-resource setting, addressing the source of the issue, and we propose a potential solution to make unsupervised NMT more robust to the lack of availability of monolingual data.

We start by giving a brief overview of the work so far in unsupervised MT in Section 2, establishing the general pipeline used to train an unsupervised system. We then identify the source of

the performance problem in LRUMT in Section 3, and propose potential improvements in Section 4. Lastly, in Section 5, we present our conclusions and lines for future work.

2 An Unsupervised MT Overview

The typical unsupervised NMT pipeline can be broken down into 3 sequential steps:

1. Train monolingual embeddings for each language
2. Align embeddings with a mapping algorithm
3. Train NMT system, initialized with aligned embeddings

In the first step, monolingual embeddings (which we will also refer to as pretrained embeddings) are most often trained in the style of WORD2VEC's skip-gram algorithm (Mikolov et al., 2013). To incorporate sub-word information, Lample et al. (2018) use FASTTEXT (Bojanowski et al., 2017), which formulates a word's embedding as the sum of its character n-gram embeddings. Artetxe (2019) uses a WORD2VEC extension PHRASE2VEC (Artetxe et al., 2018b), which learns embeddings of word n-grams up to trigrams, effectively creating embeddings for phrases.

The second step involves the alignment of the two monolingual embeddings such that the embeddings of words with identical or similar meaning across language appear close in the shared embedding space. Artetxe et al. achieve this using VECMAP (Artetxe et al., 2018a), which learns a linear transformation between the two embeddings into a shared space. If there is a large shared vocabulary between the two languages, it is also possible to concatenate the monolingual corpora and train a single embedding for both languages, effectively completing steps 1 and 2 simultaneously (Lample et al., 2018).

The third and final step is to train the NMT model. The architecture can be any encoder-decoder model, with the condition that it can translate in both directions. Models typically share an encoder and decoder for both languages, with a language token provided only to the decoder. Two objectives are used to train the model: denoising and on-the-fly back-translation. Denoising is monolingual; the model is given an altered sentence (e.g. with word order shuffling or word removal) and trained to reconstruct the original, unaltered sentence. On-the-fly back-translation involves first translating a sentence from the source language (s_{src}) to the target language (s'_{tgt}). This creates a pseudo-parallel sentence pair (s'_{tgt}, s_{src}), so the output s'_{tgt} is translated back to the source language (creating s''_{src}), and the model is trained to reconstruct the original source sentence, minimizing the difference between s''_{src} and s_{src}. Denoising and back-translation are carried out alternately during training.

The unsupervised SMT approach is fairly similar, with a replacement of step 3 (or in the hybrid approach, a step added between steps 2 and 3). In Artetxe et al. (2019) for example, a phrase-based SMT model is built by constructing a phrase table that is initialized using the aligned cross-lingual phrase embeddings, and tuning it using an unsupervised variant of the Minimum Error Rate Training (Och, 2003) method. For the hybrid model, the SMT system can then create pseudo-parallel data used to train the NMT model, alongside denoising and back-translation. In the remainder of this paper, we focus on the purely NMT approach to unsupervised MT.

3 The Role of Pretrained Embeddings in Unsupervised MT

With the pipeline established, we now turn to the LRUMT setting. In LRUMT, the existing unsupervised approaches fail somewhere along the pipeline, but simply measuring MT performance does not make it clear where this failure occurs. We speculate that the failure is relative to the quality of the pretrained word embeddings, and subsequent quality of the cross-lingual alignment. We test this hypothesis in this section.

The aligned pretrained embeddings of an unsupervised NMT system are what jump-starts the process of translation. From aligned pretrained embeddings alone, we can effectively do word-for-word translation, which is commonly measured using Bilingual Lexicon Induction (BLI). Without well-aligned pretrained embeddings, denoising and back-translation alone are not enough to produce meaningful translations.

For our following experiments[1], we train on English and German sentences from the WMT Monolingual News Crawl from years 2007 to 2017, use *newstest* 2015 for development and *newstest*

[1]Our code for running our experiments can be found at: https://github.com/Leukas/LRUMT

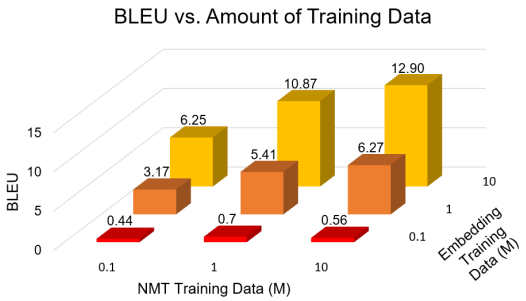

Figure 1: English→German BLEU scores of unsupervised NMT systems where the amount of training data used for the pre-trained embedding training and the amount used for the NMT model training is varied.

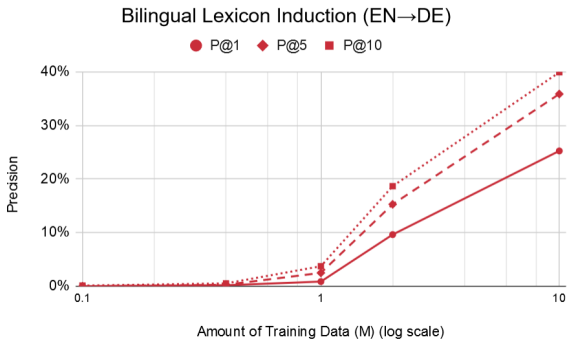

Figure 2: BLI of standard WORD2VEC using various amounts of training data, measured with precision at 1, 5, and 10.

2016 for testing, following Lample et al. (2018). The training data is filtered such that sentences that contain between 3-80 words are kept. We then truncate the corpora to sizes ranging from 0.1 to 10 million sentences per language, specified as necessary. We used UDPIPE (Straka and Straková, 2017) for tokenization[2], MOSES (Koehn et al., 2007) for truecasing, and we apply 60 thousand BPE joins (following Lample et al. (2018)) across both corpora using fastBPE.[3,4] We train the word embeddings using the WORD2VEC skipgram model, with the same hyperparameters as used in Artetxe et al. (2017b), except using an embedding dimension size of 512.[5] For embedding alignment, we use the completely unsupervised version of VECMAP with default parameters. We then train our unsupervised NMT models using Lample et al. (2018)'s implementation, using the default parameters, with the exception of 10 back-translation processors rather than 30 due to hardware limitations. We use the early stopping criterion from Lample et al. (2018).[6]

To demonstrate the importance of a large amount of training data, we vary the amount of monolingual data used for training the embeddings as well as the amount used for training the NMT system in Figure 1.[7] Even if we then use 10 million sentences per language to train the NMT system, using only 100 thousand sentences per language to train the embeddings results in a BLEU score below 1. Conversely, the NMT system can achieve a BLEU score of around 6 using embeddings trained on 10 million sentences, even when the NMT system is only trained on 100 thousand sentences per language.

We also provide Figure 2, showing the BLI scores of the aligned embeddings (using the English→German test set from Artetxe et al. (2017a)[8]) as we vary the amount of training data used for the embeddings. We can see that the BLI scores decrease dramatically as the amount of sentences decreases, matching the trend of the results from Figure 1. Although BLI has been criticized for not always correlating with downstream tasks (Glavas et al., 2019), in this case, poor alignment corresponds to poor MT performance.

In these experiments, we use VECMAP for aligning embeddings. VECMAP's algorithm begins by initializing a bilingual dictionary, which uses a word's relations to the other words in the same language, with the idea being that "apple" would be close to "pear" but far from "motorcycle" in every language, for example. However, if the quality of embeddings is poor, the random initialization of embeddings has a greater dampening effect. Using embedding similarity tasks (shown in Table 1), we find this to be the case.

We measure the quality of the monolingual embeddings using 3 similarity datasets for English:

[2]We use UDPIPE's tokenizer over the commonly used MOSES as UDPIPE learns tokenization from gold-standard labels based on the UD tokenizing standard, allowing for higher-quality dependency parsing (which will be used in Section 4).

[3]https://github.com/glample/fastBPE

[4]BPE is not applied when measuring BLI or word similarity.

[5]We use a dimension size of 512 to match the embedding size used in Lample et al. (2018)'s Transformer model.

[6]We also limit training to 24 hours. On the GPU we used to train our experiments, an Nvidia V100, limiting the training time only affected systems which used 10 million sentences per language.

[7]Although we only show results for an unsupervised NMT system, the state-of-the-art SMT systems also require initialization from pretrained embeddings. Therefore, we expect the same trend would appear.

[8]We modify the test set by truecasing it in order to match our models.

Word Similarity	Amount of Data (M)		
	0.1	1	10
EN - MEN	0.138	0.421	0.705
EN - WS353	0.018	0.461	0.628
EN - SIMLEX	0.011	0.232	0.300
DE - SIMLEX_DE	0.017	0.051	0.293

Table 1: The Spearman correlation of the similarity of word pairs (measured by cosine similarity) and human evaluation. Evaluation done using: `https://github.com/kudkudak/word-embeddings-benchmarks`

MEN (Bruni et al., 2014), WS353 (Agirre et al., 2009), and SIMLEX999 (Hill et al., 2015). We also use Multilingual SIMLEX999 (Leviant and Reichart, 2015) for German and denote this as `SIMLEX_DE`.

As we can see in Table 1, the correlation to human judgment on similarity tasks decreases dramatically as the amount of data used to train the models decreases. The poor correlation when data is limited explains VECMAP's poor alignment, as it relies on word similarity being relatively equivalent across languages for its initialization step.

4 Getting More out of Scarce Data

With the source of the problem established as the drop in quality of embeddings, we ask ourselves: how can we prevent this drop in a low-resource scenario, where considerably less monolingual data is available? We argue that the conventional word embedding methods (i.e. WORD2VEC) do not use all of the information present within sentences during the training process.

Word embedding algorithms typically define a context-target pair as a word and its neighboring words in a sentence, respectively. While this method works with a large amount of data available, it relies on the fact that a word is seen in several different contexts in order to be represented in the embedding space with respect to its meaning. When data is limited, the contexts contain too much variability to allow for a meaningful representation to be learned.

To test this, we use an embedding strategy that has a different definition of the context: dependency-based word embeddings (Levy and Goldberg, 2014). These embeddings model the syntactic similarity between words rather than semantic similarity, providing an embedding representation complementary to standard embeddings.

This section presents our findings using dependency-based embeddings (4.1). We also consider the effect of using sub-word information via FASTTEXT (4.2). With the previous two approaches, we find that ensembling models can be useful, and investigate this further (4.3). Finally, we vary context window size and report on its effect (4.4).

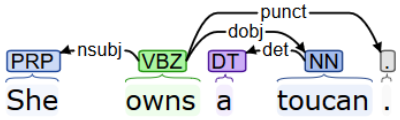

Figure 3: Example of a dependency-parsed sentence.

4.1 Dependency-Based Embeddings

Dependency parsing offers a formalization of the grammatical relationship between the words in a sentence. For each sentence, a dependency parser will create a tree in which words are connected if they have a dependency relation between them. As shown in Figure 3, the `nsubj` relation denotes the subject-to-verb relation between `she` and `owns`, for example.

Levy and Goldberg (2014) use dependency information to train word embeddings, defining the context as the parent and child relation(s) of the target word. This has two effects that distinguish dependency-based embeddings from standard embeddings. Firstly, the context is limited to syntactically-related words. For example, determiners are always limited to a context of a noun. Therefore, words of the same part-of-speech tend to be closer in the embedding space, since they have similar contexts. Secondly, the context is not limited by the distance between words in a sentence. For example, Figure 4 shows a long-range dependency between `item` and `rack`. This relation would only be captured by a standard word embedding algorithm with a large context window of length 14 or greater, whereas in the dependency-based version `rack` is one of 4 tokens in `item`'s context, and `item` is one of 6 tokens in `rack`'s context.

Levy and Goldberg (2014) also require the embedding model to predict the relation between the target word and a context word, and whether it is a parent or child relation. This explicitly trains the model to understand the syntactic relationship between two words, which provides information on the function of a word in a sentence. For example, referring back to Figure 3, the fact that `owns` has

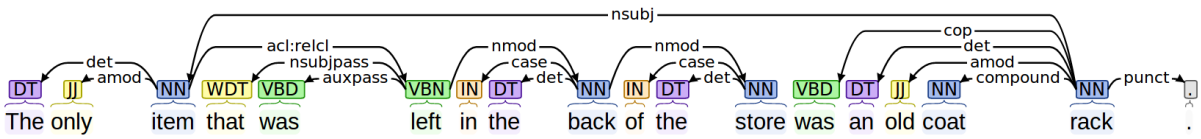

Figure 4: Example of a sentence with a long-range dependency, in this case, an `nsubj` relation between `item` and `rack`.

a `dobj` relation means that `owns` is a transitive verb. Although this information could be learned implicitly by regular WORD2VEC, as the amount of training data decreases, it becomes much harder to learn without explicit labels.

Due to their reduced context variability and their explicit learning of linguistic information, we expect dependency-based embeddings to achieve a better alignment in the low-resource setting.

In the following experiments, we use the same settings as mentioned in Section 3, apart from those explicitly mentioned. With the addition of dependency parsing into the pipeline, we apply a parser on the tokenized sentences, while truecasing is learned prior to but applied after parsing. We use the StanfordNLP parser (Qi et al., 2019), using the pretrained English and German models provided to parse our data.

Although the dependency parser that we use is supervised, therefore requiring dependency data, it is possible to train a dependency parser in an unsupervised fashion (He et al., 2018). Regardless, a dependency parser extracts linguistic information that is present in a sentence, thus our dependency-based method can still show whether using such linguistic information for training embeddings is useful for their alignment.

For training dependency-based word embeddings, we apply Levy and Goldberg (2014)'s dependency-based WORD2VEC, and compare this against the standard WORD2VEC. For the dependency-based embeddings, we use the same hyperparameters as we use for WORD2VEC.

To achieve considerable results in unsupervised NMT, it is necessary that we apply Byte-Pair Encoding (BPE) (Gage, 1994). In the dependency-based pipeline, this is learned after truecasing and applied after dependency parsing. In order to apply BPE to dependency-parsed sentences, any words that are split into multiple sub-word units will have a `bpe` relation or relations connecting them. We connected sub-word units from left-to-right, where the leftmost unit was the parent of all other units.[9]

[9]We experimented with several methods of connecting the relations, considering token length and frequency, but we found that the connection method had little impact on the resulting BLEU scores.

Amount (M)	Reg	DP	Reg+DP
0.1	0.00%	0.00%	0.00%
0.4	0.27%	0.18%	**0.62%**
1	2.49%	5.05%	**9.64%**
2	15.28%	11.32%	**18.66%**
10	35.86%	25.03%	**36.06%**

Table 2: BLI P@5 scores for aligned standard (Reg), dependency-based (DP), and hybrid (Reg+DP) WORD2VEC embeddings. The best scores are shown in bold.

In addition to the standard and dependency-based word embeddings, we also combine the two approaches, forming a hybrid embedding. This is done by training word embeddings using both methods separately with half the embedding dimension size (i.e. 256), concatenating them, and aligning them with VECMAP. We use the + symbol to denote a combined model.

Table 2 shows the BLI accuracies for the standard WORD2VEC (Reg), dependency-based WORD2VEC (DP), and hybrid (Reg+DP) embeddings as we vary the amount of monolingual sentences available to the embedding algorithms. We can see that the hybrid model outperforms the other two models at each threshold for data, apart from 100 thousand, where all three models fail entirely. Although the dependency-based model performs relatively poorly in cases where more than 1 million sentences are available, we see that the hybrid model still outperforms the regular model, which would indicate that the dependency-based model is providing complementary information to the regular model.

We also include Table 3, which shows the English→German BLEU scores[10] of our NMT systems using the pretrained standard, dependency-based, and hybrid embeddings. Here, we see that the standard embeddings outperform the other two models when they are given 2 million or more sentences to train on. We suspect

[10]We report the German→English BLEU scores in Table 8 in Appendix A.

Amount (M)	Reg	DP	Reg+DP
0.1	0.44	**0.97**	0.4
0.4	1.58	2.56	**3.26**
1	5.41	5.9	**6.99**
2	**9.31**	7.82	8.82
10	**12.9**	10.28	11.41

Table 3: English→German BLEU scores for NMT models using pretrained standard (Reg), dependency-based (DP), and hybrid (Reg+DP) embeddings. The best scores are shown in bold.

Amount (M)	Fast	Fast+Reg	Fast+DP
0.1	0.24%	0.36%	**1.45%**
0.4	0.18%	1.06%	**19.98%**
1	0.78%	**29.86%**	25.66%
2	34.09%	**35.64%**	29.98%
10	47.36%	**50.61%**	50.34%

Table 4: BLI P@5 scores for aligned FASTTEXT (Fast), and two hybrid models consisting of FASTTEXT with regular (Fast+Reg) and FASTTEXT with dependency-based (Fast+DP) WORD2VEC embeddings. The best scores are shown in bold.

this difference in performance is due to the inclusion of BPE, as that is the only difference in preprocessing. When adding the `bpe` relation to our dependency-parsed sentences, we may inadvertently isolate some sub-word units from their natural contexts. As we treat the leftmost unit as the parent, the other units will only have a relation to the leftmost unit, limiting their context and potentially adversely affecting their embedded representation.

Despite the potentially adverse effects of BPE, we see that dependency-based embeddings and hybrid embeddings outperform standard embeddings when monolingual data is limited to 1 million sentences per language or fewer.

4.2 Considering Sub-word Information

As Lample et al. (2018) and Artetxe et al. (2019) established, considering sub-word information proves very effective in increasing the performance of unsupervised MT systems. We follow Lample et al. (2018) and achieve this by using FASTTEXT. As FASTTEXT represents words as a summation of character n-grams, rarer words can have a meaningful representation if they are composed of common character n-grams. So as data becomes more scarce, FASTTEXT effectively relies on morphemes to represent words.

For FASTTEXT, we use the same hyperparameters as used for the regular WORD2VEC, apart from the context size, in which we follow Lample et al. (2018) and use a size of 5. Additionally, we create hybrid models of FASTTEXT and regular WORD2VEC concatenated (Fast+Reg), as well as FASTTEXT and dependency-based WORD2VEC concatenated (Fast+DP). The resulting BLI scores are shown in Table 4.

We can see that the inclusion of sub-word information via FASTTEXT has a very large impact on the alignment quality in general: for FASTTEXT alone, the alignment scores improve over the regular and dependency-based models, provided there are 2 million or more sentences. Unlike with regular embeddings, the Fast+DP model does not provide improvements when there are at least 1 million sentences available. With all three FASTTEXT-based models, we see a drastic improvement from 0-2% up to 20-35% when the amount of data is increased, however the Fast+DP model has this increase with less data, which may indicate that dependency information is useful in the lower resource setting.

Amount (M)	Fast	Fast+Reg	Fast+DP
0.1	0.77	**1.94**	1.16
0.4	**7.47**	7.28	5.32
1	**10.37**	9.37	7.48
2	**11.49**	11.48	10.12
10	**13.98**	13.89	11.77

Table 5: English→German BLEU scores for aligned FASTTEXT (Fast), and two hybrid models consisting of FASTTEXT with regular (Fast+Reg) and FASTTEXT with dependency-based (Fast+DP) WORD2VEC embeddings. The best scores are shown in bold.

For 100 thousand sentences, we do see some improvement, but with a P@5 of less than 2%, it is clear that none of the embedding methods tested are capable of providing embeddings of a high enough quality to allow for a decent unsupervised alignment.

While the inclusion of sub-word information via FASTTEXT outperforms the dependency-based embeddings alone, the two are not mutually exclusive: it is feasible to train a variant of FASTTEXT that uses contexts based on dependency relations to get the best of both worlds. From simple concatenation, the Fast+DP hybrid embeddings proved useful for cases where only 100-400 thousand sentences per language were available.

Table 5 shows the resulting BLEU scores for

FASTTEXT and the two previously described hybrid models.[11][12] With at least 400 thousand sentences available, we see that the non-hybrid model and the Fast+Reg hybrid perform similarly, but the Fast+DP hybrid performs worse than the other two. With only 100 thousand sentences available, both hybrid models perform better than the non-hybrid model, with Fast+Reg giving the best performance.

The BLEU scores from Table 5 as well as Table 3 seem to indicate that hybridization does not necessarily lead to better translation quality, despite often giving a higher BLI score. The BLEU score of the Fast+DP model trained on 400 thousand sentences per language stands out in particular, as the corresponding BLI score appears to indicate that the quality of the alignment should be much better than the other two models. We speculate that this could be due to one of two things: either it is due to the inclusion of BPE (as we previously discussed), or it is an artifact of VECMAP's training. Concerning the latter, VECMAP may be aligning the embeddings to the point where they are close enough for the NMT system to understand which words correspond to which, but not to the point where a large number of words will have their corresponding words in the other language close enough to be counted for the BLI precision at 5 score. Therefore, the large jump in BLI scores can be misleading in terms of alignment quality for unsupervised NMT.

Overall, the performance of FASTTEXT indicates that the use of sub-word information is very important to the performance of the NMT system, as we see both BLI and BLEU score improvements when comparing FASTTEXT to standard WORD2VEC. Along with the performance of the dependency-based embeddings, this supports the idea that linguistic information as a whole can be useful in improving translation quality in unsupervised NMT.

[11]We report the German→English BLEU scores in Table 9 in Appendix A.

[12]The BLEU scores are not directly comparable to the results of Lample et al. (2018) for a couple of reasons (apart from the hardware limitation previously mentioned): 1. We use VECMAP to align embeddings, whereas they concatenate corpora and train a singular embedding. 2. We use a maximum of 10 million sentences per language, they use the entire WMT News Crawl dataset, which is well over 100 million sentences per language.

4.3 Ensembling of Embeddings

As our hybrid embeddings have shown to have an increase in performance, we note that this could be due to the effect of ensembling two embeddings with different random weight initializations rather than due to the differences between the embedding algorithms. To test this, we train two embeddings using the same algorithm (but different weight initializations) and concatenate them in the same manner as the hybrid models. Using this method, we produce Reg+Reg, DP+DP, and Fast+Fast, and we compare them to our hybrid models in Table 6.

The scores show that the improvement found in Reg+DP is greater than the improvement found by ensembling either of its two constituent models. This indicates that there is a complementary relationship between regular and dependency-based WORD2VEC. As for Fast+Fast, the model performs better than the two hybrid models using FASTTEXT when the number of sentences ranges from 400 thousand to 2 million, with the greatest improvement found at 400 thousand sentences per language. While there is a greater improvement from Fast+Fast compared to Fast+Reg and Fast+DP, this may be more due to the poor quality of the Reg and DP components of the hybrid models, whose contribution may be hindering the alignment rather than helping. Overall, ensembling 2 embeddings from the same embedding algorithm yields marginal improvements in alignment quality, whereas ensembling 2 embeddings from different algorithms can potentially yield greater benefits.

4.4 Context Size

Seeing as the context plays a role in the alignment quality of embeddings, we vary the context window size of WORD2VEC and FASTTEXT embeddings to see its effect. Additionally, using a context size of 1 with WORD2VEC produces embeddings which are better suited for inducing part-of-speech tags (Lin et al., 2015), which could also aid with alignment. As such we test on context sizes of 1, 3, 5, and 10.

The results overwhelmingly indicate that a larger context size is better for alignment when there are at least 1 million sentences per language available. This may explain why the dependency-based embeddings do not perform well relative to the standard WORD2VEC and FASTTEXT embeddings. In the sentence in Figure 4, for example, the

Amount (M)	Reg+Reg	DP+DP	Reg+DP	Fast+Fast	Fast+Reg	Fast+DP
0.1	0.00%	0.00%	0.00%	0.84%	0.36%	**1.45%**
0.4	0.09%	0.44%	**0.62%**	**24.14%**	1.06%	19.98%
1	6.07%	4.67%	**9.64%**	**31.26%**	29.86%	25.66%
2	15.50%	11.46%	**18.66%**	**35.86%**	35.64%	29.98%
10	35.93%	25.30%	**36.06%**	47.16%	**50.61%**	50.34%

Table 6: BLI comparison of ensemble models (Reg+Reg, DP+DP, and Fast+Fast), to the aforementioned hybrid models (Reg+DP, Fast+Reg, and Fast+DP).

Amount (M)	WORD2VEC				FASTTEXT			
	1	3	5	10	1	3	5	10
0.1	0.00%	**0.12%**	0.00%	0.00%	0.12%	**0.60%**	0.24%	0.00%
0.4	0.00%	0.00%	0.00%	**0.27%**	0.18%	0.27%	0.18%	**0.35%**
1	0.00%	0.08%	1.48%	**2.49%**	0.00%	0.23%	0.78%	**28.07%**
2	3.16%	5.66%	13.15%	**15.28%**	23.14%	32.33%	34.09%	**35.05%**
10	27.06%	32.27%	33.90%	**35.86%**	39.92%	45.20%	47.36%	**48.58%**

Table 7: BLI P@5 scores for aligned FASTTEXT, and WORD2VEC, with varying window sizes of 1, 3, 5, and 10.

largest context is 6 for the word `rack`, and the average context size is 1.83. Given the increases we see from WORD2VEC and FASTTEXT with a larger context size, it is likely we will see a large increase in alignment quality for dependency-based embeddings as well if they can be trained with a larger context.

5 Conclusion and Future Work

Unsupervised NMT has made great strides in making MT more accessible for language pairs that lack parallel corpora. We attempt to further this accessibility by introducing LRUMT, where monolingual data is also limited. Our results show that, in the current state-of-the-art pipeline, the quality of the pretrained word embeddings is the main issue, and that using syntactically-motivated dependency-based embeddings has the potential to improve performance when monolingual data is limited.

We also see that the inclusion of sub-word information for training word embeddings provides a crucial performance increase, which provides further evidence that using the latent linguistic information in a sentence can improve embedding alignment quality.

Finally, on the topic of context size, we find that a larger context size is almost always better, most noticeably when more data is available. This helps explain the poorer performance of the dependency-based embeddings on larger amounts of data.

To improve upon dependency-based embeddings for unsupervised NMT, we consider two avenues to explore: including sub-word information and increasing the context size. To include sub-word information, it should be possible to combine the training methods of FASTTEXT and dependency-based WORD2VEC. To increase the context size, one might consider including a word's grandparent, grandchildren, and siblings (its parent's other children) as part of the context.

We also note that we currently use a pretrained dependency parser, trained on labelled dependency data, which is often harder to come by than parallel data. We plan to switch to using unsupervised dependency parsing techniques to ensure this method is accessible for all languages.

Furthermore, there are several potential methods for incorporating more linguistic information into embeddings. One such possibility would be to use a morphological segmenter such as MORFESSOR (Virpioja et al., 2013) rather than BPE, which would likely provide better results for more morphologically-rich languages. As we only test on English–German, our future work will test this new paradigm on other language pairs, particularly those in which unsupervised NMT fails to perform such as English into morphologically-rich languages.

References

Agirre, Eneko, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. 2009.

A study on similarity and relatedness using distributional and wordnet-based approaches.

Artetxe, Mikel, Gorka Labaka, and Eneko Agirre. 2017a. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

Artetxe, Mikel, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017b. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*.

Artetxe, Mikel, Gorka Labaka, and Eneko Agirre. 2018a. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798.

Artetxe, Mikel, Gorka Labaka, and Eneko Agirre. 2018b. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, November. Association for Computational Linguistics.

Artetxe, Mikel, Gorka Labaka, and Eneko Agirre. 2019. An effective approach to unsupervised machine translation. *arXiv preprint arXiv:1902.01313*.

Barrault, Loïc, Ondřej Bojar, Marta R Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. 2019. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bruni, Elia, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

Gage, Philip. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Glavas, Goran, Robert Litschko, Sebastian Ruder, and Ivan Vulic. 2019. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. *arXiv preprint arXiv:1902.00508*.

Hassan, Hany, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

He, Junxian, Graham Neubig, and Taylor Berg-Kirkpatrick. 2018. Unsupervised learning of syntactic structure with invertible neural projections. *arXiv preprint arXiv:1808.09111*.

Hill, Felix, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Lample, Guillaume, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Lample, Guillaume, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. *arXiv preprint arXiv:1804.07755*.

Leviant, Ira and Roi Reichart. 2015. Separated by an un-common language: Towards judgment language informed vector space modeling. *arXiv preprint arXiv:1508.00106*.

Levy, Omer and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 302–308.

Lin, Chu-Cheng, Waleed Ammar, Chris Dyer, and Lori Levin. 2015. Unsupervised pos induction with word embeddings. *arXiv preprint arXiv:1503.06760*.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Qi, Peng, Timothy Dozat, Yuhao Zhang, and Christopher D Manning. 2019. Universal dependency parsing from scratch. *arXiv preprint arXiv:1901.10457*.

Straka, Milan and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada, August. Association for Computational Linguistics.

Virpioja, Sami, Peter Smit, Stig-Arne Grönroos, Mikko Kurimo, et al. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline.

A German→English Results

We report the BLEU scores for German→English in Tables 8 and 9. Comparing these BLEU scores to the respective English→German BLEU scores in Tables 3 and 5, we see that the best performing models are the same for both translation directions. This suggests that the translation direction is not important for evaluating the relative differences unsupervised NMT systems. However, since English and German are related languages, this could also simply be a feature of this language pair.

Amount (M)	Reg	DP	Reg+DP
0.1	0.54	**1.20**	0.57
0.4	1.95	2.91	**3.71**
1	6.99	7.14	**8.74**
2	**11.90**	10.03	11.44
10	**16.97**	12.95	15.07

Table 8: German→English BLEU scores for NMT models using pretrained standard (Reg), dependency-based (DP), and hybrid (Reg+DP) embeddings. The best scores are shown in bold.

Amount (M)	Fast	Fast+Reg	Fast+DP
0.1	1.11	**2.39**	1.35
0.4	**10.01**	9.98	7.10
1	**13.68**	12.38	9.99
2	**15.27**	14.82	13.15
10	**18.40**	18.31	15.16

Table 9: German→English BLEU scores for aligned FASTTEXT (Fast), and two hybrid models consisting of FASTTEXT with regular (Fast+Reg) and FASTTEXT with dependency-based (Fast+DP) WORD2VEC embeddings. The best scores are shown in bold.