Limburgish-Dutch Translation: Case study of Unsupervised Neural Machine Translation

T. Debets, S. Jain, J. Kunnen, K. Shcherbakov, K. Stessen

***Abstract*—Machine translation has recently achieved impressive performance-gains due to advances in deep learning and large parallel corpora. These resulted in almost human-level translation. In an effort to achieve similar result for low-resource languages, that do not have large parallel corpora, researchers have explored the potential of unsupervised machine translation. For this project we gathered a data-set of both Dutch and Limburgish in similar domains before training a Cross-lingual Language Model. It was able to achieve a BLEU score of 31.73 for Limburgish to Dutch, and a 20.21 BLEU score for Dutch to Limburgish.**

I. Introduction

Neural Machine Translation (NMT) is one of several research fields within Natural Language Processing (NLP) besides others such as classification and natural language inference [1] [2] [3]. Machine translation can be supervised or unsupervised. With supervised NMT, parallel data is required, which is not always readily available. Therefore, in the more recent years, research has shifted towards unsupervised NMT [4] [5]. At first, languages with large text corpora, have been investigated and used for unsupervised machine translation, i.e. English to German. Following the same shift as from supervised to unsupervised, the translation models have been focusing more on the sparse languages. In [6] research has been done in Pidgin English. This language is estimated to be spoken by around 80 Million people. Ogueji et al. developed an unsupervised neural machine translation model between Pidgin English and English.

In the Netherlands and Belgium there is a region where Limburgish, a local dialect, is spoken. It is fairly similar to Dutch and German, but still reasonably distinct from those two languages. The language itself is spoken by around 2 million people. Therefore, no research has been done for this language regarding translators.

Limburgish is no homogeneous language. Every town has its variant of the language [7]. In an article [8] published on a Limburgish news site, de Limburger, a Dutch doctor explains the difficulties she has while treating Limburgish speaking patients. She emphasizes the importance to understand every word a patient speaks while explaining symptoms. A misunderstood expression could lead to a misdiagnosis and thus ineffective or even harmful treatment.

A Limburgish - Dutch translator could be a solution to similar situations. The limited number of users and the fact that it is mainly a spoken language resulted in limited written sources. Therefore, the following research questions guide our research:

- How can a data base for Limburgish be created?
- Which architectures and methods do exist to build an unsupervised Neural Machine Translator?

This article is structured as follows. Section II gives an overview of the related work in this field. Subsequently, Section III and IV explore the methods used to build a database and which architectures have been researched. Sections V and VI discus the experimental setup and the results respectively. Finally, the last section, Section VII, contains the conclusion and a discussion on potential future work.

II. Related work

In recent years, several researchers [9] [10] investigated language modeling for pretraining Transformer encoders. Applying these approaches lead to drastic improvements on various classification tasks. Ramachandran et al. [11] show that using a pretrained language model can provide improvements on machine translation tasks, even for high-resource language pairs.

In 2013, Mikolov et al. [12] showed a method that leverages small dictionaries to align word embeddings from several languages. Follow-up studies proved that cross-lingual representations improve the quality of monolingual representations even further [13]. Continuing on this research, Xing et al. [14] show that orthogonal transformations are sufficient to align these worth distributions. Smith et al. [15] indicated that the need for cross-lingual supervision can be further reduced, until Conneau et al. [16] removed it completely. The work of Lample and Conneau [5] take these ideas one step further by aligning the text distributions and reducing the need for parallel data. Ruder et al. [17] made an overview of cross-lingual embedding learning methods.

Johnson et al. [18] show that one single sequence-to-sequence model can be used to perform machine translation for several language pairs, by using a single LSTM encoder and decoder. This work is used by Artetxe and Schwenk [19] to show that the encoder can be used for producing cross-lingual sentence embeddings. All these methods require a significant amount of parallel data, Lample et al. [20] and Artetxe et al. [4] both show a completely unsupervised way to align sentence representations. Lample and Conneau [5] designed a model that uses these previous mentioned researches to develop an unsupervised neural machine translator by using language modelling and iterative back-translation.

III. Data

The model will be trained in an unsupervised fashion, and therefore, the training data will not require labeling and solely consist of pre-processed sentences. Creating an Unsupervised Neural Machine Translator capable of translating from Dutch to the more region-specific Limburgish requires two well-balanced Data sets, a Dutch and Limburgish one, to train the model extensively. As mentioned in the introduction, Limburgish is mainly a spoken language and therefore, readily available data is hard to find. Hence, a database had to be built from scratch. In addition, we also decided to build a new Dutch database, despite having access to existing ones. One advantage is that the sources are known. Secondly, the contents of the set will be known, which will be imported to analyse the results of our translator.

For the acquisition of data for the Dutch data set, sentences from the Dutch-Wikipedia were extracted. For this a python library has been used, WikiExtractor [21]. This tool extracts and cleans text from a Wikipedia database dump, which led to the retrieval of more than more than 2.000.000 articles which contained almost 18.9 million sentences.

For Limburgish significantly fewer records were available online, compared to the vast amounts of data available in Dutch. Fortunately, there exists a Limburgish Wikipedia, which has been used as our main resource for acquiring training data. Next to using Wikipedia as a source, data was also extracted from other Limburgish websites. Finally, writers, enthusiasts and organisations that promote Limburgish culture have been contacted who provided us with approximately 8.000 sentences. In total we were able to build a data set that contains almost 275 thousand Limburgish sentences. The sources can be found in Appendix A.

A. Analysis

From the start of this project, there was concern about the lack of a unified spelling in Limburgish. It is entirely a spoken language and the written sources are a phonetic approximation. Therefore, it seemed plausible that words might have a multiple different spellings. This poses the obvious risk of semantic loss and as a result a less performant model. In order to test this hypothesis, all not infrequent words were compared on a hybrid similarity metric based on both the Levenshtein distance and the 2-gram overlap between the Kölner Phonetik encodings. Kölner Phonetik was selected due to the similar pronunciation in Limburgish and German. For our new metric a cut-off point of 0.8 was used which resulted in 7538 word-pairs. A small test (250 instances) using the expertise of multiple native speakers resulted in an accuracy of approximately 58 percent. Some examples are:

1) gemeinte, gemeindje and gemeint
2) hebbe, höbbe and hubbe
3) stad and sjtad
4) ierste and eerste

Setting a higher cut-off point does not seem to be sufficient to increase the precision. Therefore an automated pre-processing to reach an uniform spelling is unfeasible. This is a challenge in itself and an interesting subject for further research.

Sentences	**Limburgish**	**Dutch**
Total Training Set	274.315	18.899.914
Matched Training Set	274.315	5.535.535

TABLE I: The amount of sentences in the complete and the matched training sets.

B. Matching

As can be seen in Table I in the 'Total Training Set' column, there is a vast difference between the gathered number of Limburgish and Dutch sentences. In both papers of Yunsu Kim [22] and Lukan Edman [23], the researches investigate big differences in the amount of data in two sets for unsupervised machine translation. The results of both papers show that the amount of sentences in the data sets do not have a significant influence on the performance of the translators they used. They found however that the domain similarity of the two languages is highly influential.

With this information, a second Training set for the Dutch data set was created. For this training set, domain similarity between the Dutch and Limburgish data set is prioritized. Experiments have been conducted for both sets to see if the results are similar as in the papers mentioned above. See the Results section, Section VI, for the outcomes of these experiments.

Both corpora consisted mainly out of Wikipedia articles (Limburgish: 80% and Dutch: 100%). After closer investigating of the sentences, using the existing knowledge of the language, it was found that often the vowels differed while the consonants were similar. This knowledge was then used to achieve a similar domain in both languages. Articles were matched based on the consonants in their titles. Note that this matching process is not perfect. There are more differences in the two languages than only their use of vowels and words with the same consonants in both languages can have different meanings. In addition, it is possible due to the non-homogeneity of the Limburgish language, that several of the Limburgish articles refer to the same Dutch Wikipedia article. As a result of time constraints and the size of the data sets, this imperfect matching can arguably still give us better results than using the complete Dutch data set. As you can see in Table I, the Dutch data set has already been compressed to a little more than 5.5 million sentences after matching. The big difference in the number of sentences that still remain in the Matched Training set, is due to the length of the articles on Wikipedia. In Dutch most articles elaborate more on the topic than their Limburgish counterpart. It is no exception for the Dutch articles to be more than ten times as long. A word-frequency analysis of the selected corpora was performed to evaluate the domain similarity. The Limburgish articles have a strong focus on local history, culture and language. After the matching the same seems to be true for the dutch corpus based on the most frequent words.

Sentences	Limburgish	Dutch
Short Validation Set	1141	1141
Short Test Set	1226	1226
Long Validation Set	2044	2044
Long Test Set	2000	2000

TABLE II: The amount of sentences for the validation and test sets.

C. Test and Validation Sets

In addition to the unparalleled training data for Dutch and Limburgish, sentences are gathered that are a direct translation of each other to test our model. Table II shows the exact number of sentences in each set. As an experiment shorter and longer versions of the test and validation set have been created. The long validation and test sets consist of the short validation and test set with additional sentences. The results of both the short and long set can be found in the Results section, Section VI.

IV. Models

To build an Unsupervised Neural Machine Translator, research has been conducted for two models, the Pidgin English model [6] and Facebook's cross-lingual language model (XLM) [5]. This section describes both.

A. Pidgin UNMT

The authors trained cross-lingual embedding via monolingual mapping. In this method a linear method is learned between two pretrained monolingual word embeddings. In their research, the authors tested two methods, supervised and unsupervised cross-lingual embedding alignment. The model, inspired by [24], used for UNMT existed of a Transformer [25] with 10 attention heads. In the Transformer, there are 4 encoder and 4 decoder layers with 3 encoder and decoder layers shared across languages. For the decoder to perform well, its inputs should be produced by the encoder it is trained with or the input comes from a similar distribution as that of the encoder. Therefore, the authors made sure that the encoder encodes sentences from both the source and target language to the same latent space.

This enforcement has been done by adversarial training following Lample et al. [20]. A discriminator is trained to classify encodings of source and target sentences. The encoder is trained to mislead the discriminator to make sure the latent space representations for both the source and target language sentences are indistinguishable. Furthermore, the latent space representation is used for both the language model and the translation task. Therefore, the representation of the language model is transferred to the translation task.

At each training step, the model performs the following steps:

1) Discriminator training to predict the language of an encoded sentence.
2) Denoising autoencoder training on each language (this is similar to training a language model to learn useful patterns for reconstruction).

Algorithm for our Unsupervised Neural Machine Translation

1: **procedure** Training(D_{src}, D_{tgt}, enc, dec, discr, N)
2: $M_{src-tgt}^{(0)}$ and $M_{tgt-src}^{(0)}$ initialized models for translation
3: **for** i =1 **to** N, **do**:
4: **discriminator training** and **language model training** for src and sgt
$\theta_{discr}, Z \leftarrow \arg\min L_{discr}$;
$\theta_{enc}, \theta_{dec}, Z \leftarrow \arg\min L_{auto}$;
$\theta_{enc}, Z \leftarrow \arg\min L_{adv}$;
5: **back-translation training**
- given LM_{src} *and* LM_{tgt} as language models trained in step 4;
- generate translations using current models $M_{src-tgt}^{(i-1)}$ & $M_{tgt-src}^{(i-1)}$, leveraging LM_{src} *and* LM_{tgt};
- train new translation models $M_{tgt-src}^{(i)}$ & $M_{src-tgt}^{(i)}$ to reconstruct original sentences from generated translations, leveraging LM_{src} *and* LM_{tgt}
6: **end**

Fig. 1: Pidgin Algorithm [6]

3) On-the-fly back translation of a given sentence.

Figure 1 shows the steps of the algorithm.

B. Cross-lingual Language Model (XLM)

The model introduced by Lample and Conneau [5] uses a same shared vocabulary for all languages, establishing a common embedding space for tokens from all languages. Hence, languages using the same script or languages that have similar words map better to the common embedding space. The tokenization of the corpora is done using Byte-Pair Encoding (BPE).

There are three methods discussed in the paper, to train the XLM: Causal Language Modelling, Masked Language Modeling and Translation Language Modeling.

1) Causal Language Modeling (CLM): In this language model, the objective is to maximize the probability of a token x_t to appear at the position 't' in the sequence, given all the tokens $x_{<t}$ (all the tokens preceding the t^{th} token) in the same sequence. Equation 1 shows the maximization objective. This model is based on the paper of Dai et al. [26].

$$\max_{\theta} \log p_\theta(x) = \sum_{t=1}^{T} \log p_\theta(x_t|x_{<t}) \tag{1}$$

2) Masked Language Modeling (MLM): This type of language modelling incorporates denoising autoencoding. The objective is to maximize the probability of a masked token x_t which is at position 't' in the sequence, provided all the tokens in the same sequence, $\hat{x}$. Equation 2 refers to the objective function, unfortunately, there is no explanation of the parameter m_t. This model is based on the paper of Dai et al. [26] and Devlin et al. [27].

$$\max_{\theta} \log p_\theta(\bar{x}|\hat{x}) \approx \sum_{t=1}^{T} m_t \log p_\theta(x_t|\hat{x}) \tag{2}$$

To illustrate better, consider the sentence *"During monsoon, it _____ heavily and floods the region"*. Humans can predict that the word *"rains"* fits the blank position in the provided

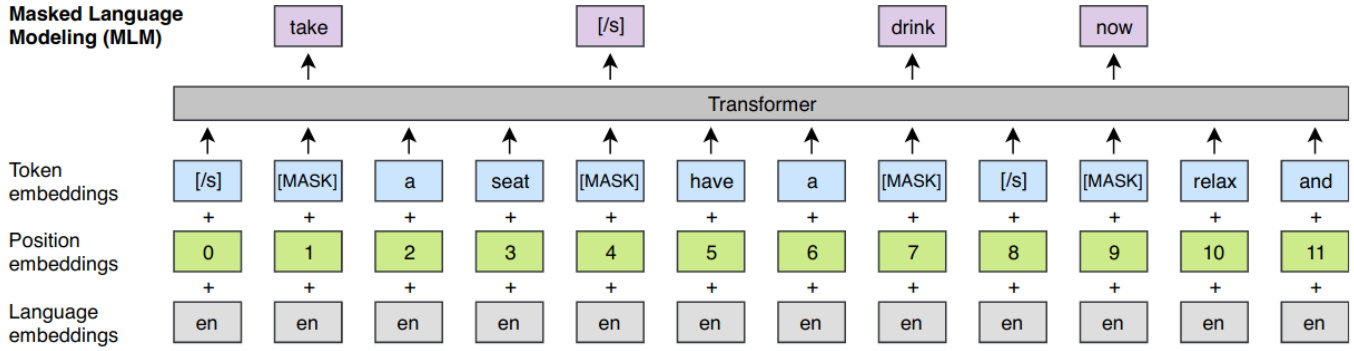

Fig. 2: Masked Language Model [5]

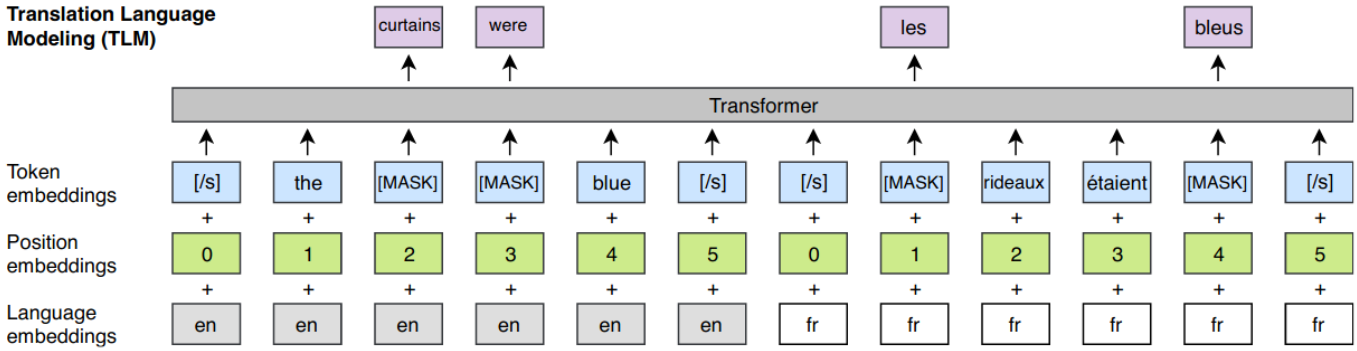

Fig. 3: Translation Language Model [5]

sentence, based on the context and general knowledge. BERT may not know what *heavily*, *monsoon* are, but it finds the linguistic patterns and the context these words are generally used in, and makes the most likely prediction. Thus, basically a masked/incomplete sentence is provided to BERT and it is asked to make the prediction for the masked word. Figure 2 shows the MLM. Note that there is a difference between BERT's and the MLM's approach. BERT uses pairs of sentences, whereas MLM uses streams of arbitrary number of sentences and truncate once the length is 256.

3) Translation Language Modeling (TLM): Both the CLM and MLM model work on monolingual corpora, these are unsupervised models. Hence, the authors propose a method, the Translation Language Model, that takes advantage of available parallel translation data. This model is an extension of the MLM. A sequence of parallel sentences is fed to the BERT model, by randomly masking tokens in both the source and the target sentences. Figure 3 shows this principal. All the words in the sequence contribute to the the prediction of the given masked word, hence establishing a cross-lingual mapping among the tokens.

4) Cross-lingual language model pretraining: The models mentioned in the previous sections can all be used to leverage for downstream tasks. The authors mention the following five tasks:

1) Zero-shot Cross-lingual Classification. On top of the XLM a classification layer is added and is trained on the English NLI data set. Afterwards, the model is evaluated on 15 XNLI languages. Since the model has not been tuned to classify sentences from these 15 languages, it is a zero-shot learning example.
2) Unsupervised Neural Machine Translation. The authors propose pre-training a complete encoder-decoder architecture with a cross-lingual language modeling objective.
3) Supervised Neural Machine Translation. The encoder and decoder are loaded with pretrained weights from XLM and then fine-tuned over the supervised translation data set.
4) Low-resource Language Modeling. For this task, the aforementioned similarity between scripts or similar words, provide a better mapping.
5) Unsupervised Cross-lingual Word Embeddings. The model has a shared vocabulary, so the lookup table of the XLM model gives the cross-lingual word embeddings.

V. Experiments

In Section IV an overview of the researched models has been given. This section will discuss the experiments done related to these models. All the experiments have been run on Google Colab Pro, with the use of a single GPU when available to speed up the experiments.

Experiment	Data set	Validation set	Test set
Experiment 1	5 million sentences WMT'07 + WMT'08 German data set + 5 million sentences WMT'07 + WMT'08 English data set	WMT'13 parallel data set	WMT'16 parallel data set
Experiment 2	complete Limburgish + complete Dutch data set	Smaller validation set	Smaller test set
Experiment 3	complete Limburgish + matched Dutch data set	Smaller validation set	Smaller test set
Experiment 4	complete Limburgish + matched Dutch data set	Larger validation set	Larger test set

TABLE III: Conducted experiments

A. Pidgin UNMT

As described before, this model was tested in a supervised and unsupervised manner. Therefore, experiments are conducted for this model. The Pidgin model can be found on Github[1]. Unfortunately, while testing this model both in a supervised and unsupervised way. The unsupervised method described by the others, only had a precision score of 0.0332. In the meanwhile, Supervised Alignment with a Retrieval Criterion, scored 0.1282, resulting in a BLEU score of 7.93 from Pidgin to English and a BLEU score of 5.18 for the other way around. Therefore, on their repository, the authors only published the supervised method. To be able to run this model, a lexicon, a bilingual dictionary is required. Sadly, for our research, such a dictionary is not available. In the end, this model could not be experimented on.

B. Cross-lingual Language Model

For our experiments, the model used can be found on Github[2] and is adjusted accordingly. Since there is no parallel data available in our case, the TLM model is immediately discarded. According to the paper [5], the MLM model performs better than the CLM model. Therefore, only experiments have been conducted on the MLM model.

The parameters used in our experiments are the standard parameters mentioned in their repository, with the exception of one. Owing to the limited time, the number of BPE tokens have been reduced to only 10.000 to make the model less complicated. Table III shows the conducted experiments. Experiment 1 is to establish a base line for the other experiments. The MLM model will train for 10 hours or if the perplexity of the validation set will not improve for 25 epochs. The XLM model will train for 10 hours as well or if the BLEU score of the validation set will not improve for 10 epochs.

VI. Results & Discussion

The results of the experiments are presented in Table IV. This table shows all the results of Limburgish to Dutch and Dutch to Limburgish for both the validation and test set. One can see that Experiment 3 with a matched Dutch training set is superior to the experiment with a regular training data set, where no matching was performed, which is expressed in a higher BLEU score. Also, one can notice that in the experiment with matched Dutch training data, perplexity is several times lower compared to the experiment on the full Dutch data set. Also, the results obtained show and prove that the domain similarity of the two languages highly influence the performance of the model which is reflected in the higher BLEU score of the model trained on the matched data (Experiment 3) compared to the model trained on the full data set (Experiment 2). Another reason can be that the difference in the amount of training data is too big, which can lead to overtraining. This is the results that Edman et al. [23] concluded from their research.

Furthermore, when increasing the validation and test sizes, this has minimal influences in our case, as can be seen in the results of Experiment 4. For this test, the same training data is used as for Experiment 3. This means that increasing the test and validation size in this case is not beneficial, or thus not have influence in the model itself. The results can be of slightly worse training of the model. On closer inspection of the results, it is noticeable that Limburgish to Dutch seems easier for the model than from Dutch to Limburgish.

Unfortunately, Experiment 1 (German-English) was not conducted in a correct manner due to time constraints. The results that are achieved had a BLEU score of approximately 8. To achieve this, the Masked Language Model is trained for approximately 10 hours. The training of the Translator ran for approximately 6 hours. Comparing it with the results of the conducted experiments in this research, it shows that, translating from Limburgish to Dutch and the other way around is easier for the machine. The authors of the Facebook's XLM model already gave a possible explanation: Low-resource Language Modeling. For this task, the similarity between scripts or similar words, provide a better mapping. Although Limburgish is not a homogeneous language, the script is the same, the sentence structure is similar and often words do lookalike.

The non-homogenity of the language can be the reason for the UNMT, Dutch to Limburgish can be more difficult. Words in Maastricht can be written different from words in Venlo (South of Limburg versus North of Limburg). The variant of the dialect has not been specified during the training and testing of the model, therefore, the model does not know which version of Limburgish it has to translate too, which causes the model to perform worse than translating from Limburgish to Dutch.

Comparing the achieved results with the results of the original paper of the XLM model [5], it shows that our results are surprisingly good. Their model trained for two weeks and achieved for German-English a BLEU score of 34.3, while our best score is for Limburgish-Dutch and achieved a BLEU score of 31.73 when trained for approximately 10 hours for the MLM model and achieving the best translation already after

[1]https://github.com/keleog/PidginUNMT

[2]https://github.com/facebookresearch/XLM

	Validation set					
	Limburgish-Dutch			Dutch-Limburgish		
	ppl	acc	BLEU	ppl	acc	BLEU
Experiment 2	53.29	59.97	31.31	48.70	59.38	22.25
Experiment 3	20.33	67.46	37.25	28.26	61.00	25.11
Experiment 4	25.56	65.25	35.53	31.52	59.25	24.19
	Test set					
	Limburgish-Dutch			Dutch-Limburgish		
	ppl	acc	BLEU	ppl	acc	BLEU
Experiment 2	77.45	55.62	25.39	73.39	53.69	18.21
Experiment 3	35.55	62.19	31.73	43.92	55.43	20.21
Experiment 4	43.31	60.48	30.85	47.46	54.34	20.43

TABLE IV: The table shows the perplexity, accuracy and BLEU score for Limburgish-Dutch and Dutch-Limburgish for both the validation and test set. The results of Experiment 1 is missing due to loss of results and time constraints. The achieved results of Experiment 1 had a BLEU score of approximately 8.

Dutch Original	Dutch Complete	Dutch Matched
2020 was een bijzonder jaar, ook voor Veldeke	2020 was een bijzonder jaar , ook voor Veluwe	2020 was een bijzonder jaar, ook voor Veldeke
Meertaligheid is een verschijnsel van alle tijden	Mietas is een verschijnsel van alle tijden	Menselijke ontwikkeling is een verschijnsel van alle tijden
Na 2013 is gekozen voor een ietwat andere opzet	Na 201ophoridae is besloten voor ' een andere manier	Na 2013 is gekozen voor ' een andere manier '
Zo zijn er tal van kerstliedjes in het Limburgs bekend	Aanvankelijk zijn d 'r vanno Spanjaarl Limburgse kersleedjes	Zo zijn d ' r väöl Limburgse kersliedjes
Hij is 24 jaar en is docent in Roermond	Hij is 24 jaar en wJohrk als docent in Remund	Hij is 24 jaar en wél als docent in Remund

TABLE V: Textual results Dutch

Limburgish Original	Limburgish Complete	Limburgish Matched
2020 waor e bezunder jaor, ouch veur Veldeke	2020 waor e bezunder jaor , ouch veur Veldeke	2020 waor e bezoonder jaor, ouch veur Veldeke
Mietaolegheid is e versjijnsel vaan alle tije	Meertaligheid ies n versjiensel van alle tieje	Meertaligheid is n versjiensel van alle tieje
Nao 2013 is gekaoze veur 'ne andere meneer	Nao 201v-2012 ies gekoze veur n ietwat ander opzet	Nao 2014 is gekoze veur n ietwat aander opzat
Zoa zeen d'r vääl Limburgse kersleedjes	Zo zeen dr tal van kerkelike tekste in t Limburgs bekend	Zoa zint dr tal van kerstleedsjes in t Limburgs bekend
Heer is 24 jaor en wèrk es docent in Remund	Hae ies 24 jaor en ies docent in Roermond	Hae is 24 jaor en is docent in Roermond

TABLE VI: Textual results Limburgish

11 epochs. Our results show that for similar languages, and Low-resource Language Modeling, the XLM model performs well.

Another observation made in our experiments is that the reduction in BPE codes, lowered the perplexity score which reaffirms the understanding that when there are lesser BPE codes, the model is less confused. Lastly, the BLEU score could have been higher, if access to more number of GPUs was available. This would allow larger batches. Our understanding is strengthened by a documentation [28] regarding multiple GPUs that mentions "Larger number of GPUs lead to larger minibatch sizes, thus increasing training efficiency".

A. Textual Results

To give more insight in the results our model achieved, some sentences of the Dutch test set have been printed in Table V and of the Limburgish test set in Table VI. It is noticeable that the model performs better in translating from Limburgish to Dutch than Dutch to Limburgish. This can be seen in the number of mistakes the model makes in its word choice. This is probably due to the fact that the Limburgish database is built out of dialects from different regions. Another observation is that the model performs better for the Dutch Matched data in comparison to the complete Dutch database. This is evident from the use of words of the model, for example in the third Dutch sentence "Na 2013 is gekozen voor een ietwat andere opzet" it is very clear that the model translated the sentence better for the Matched set than the complete set.

VII. Conclusion & Future work

In this article, an Unsupervised Neural Machine Translator (UNMT) has been built to translate Limburgish to Dutch and vice versa. Firstly, data were gathered for both Dutch and Limburgish, before being analyzed. For our experiments, Limburgish Wikipedia articles were matched with Dutch articles to create a different data set, considering the importance of domain similarity.

The models researched to build a UNMT are: Model developed for Pidgin English to English and Facebook's Cross-lingual Language Model (XLM). Although Pidgin English model has been mentioned as an unsupervised translator, a bi-lingual dictionary is required for building it. Therefore, this model has been discarded during our experiment. In case of the XLM model, the Masked Language Model is used as the pre-trained language model for our UNMT.

Based on our experiment, domain similarity is of importance when dealing with low-resource Language Modeling. Another aspect is that the model can overtrain if the difference in the amount of data is too large. Using the domain similarity data set for the XLM model resulted in a BLEU score of 31.73 for Limburgish to Dutch, and a 20.21 BLEU score has been achieved for Dutch to Limburgish, showing that the

non-homogeneity of Limburgish plays a factor in Translating from Dutch to Limburgish. Furthermore, the XLM model can perform well for similar language pairs.

In future work, a proper experiment with English-German languages may be conducted for quantitative comparison with the model in this paper. Also, the hyperparameters search can be investigated more thoroughly in the future in order to improve the model performance. Furthermore, the importance of domain similarity, overtraining of the model due to excessive data in one language can be investigated. Finally, the importance of the GPUs can be determined by training a model with several GPUs.

REFERENCES

[1] M. M. Lopez and J. Kalita, "Deep learning applied to NLP," *CoRR*, vol. abs/1703.03091, 2017. [Online]. Available: http://arxiv.org/abs/1703.03091

[2] J. Zhang and C. Zong, "Deep neural networks in machine translation: An overview," *IEEE Intell. Syst.*, 2015.

[3] S. P. Singh, A. Kumar, H. Darbari, L. Singh, A. Rastogi, and S. Jain, "Machine translation using deep learning: An overview," in *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, 2017, pp. 162–167.

[4] M. Artetxe, G. Labaka, E. Agirre, and K. Cho, "Unsupervised neural machine translation," *CoRR*, vol. abs/1710.11041, 2017. [Online]. Available: http://arxiv.org/abs/1710.11041

[5] G. Lample and A. Conneau, "Cross-lingual language model pretraining," *CoRR*, vol. abs/1901.07291, 2019. [Online]. Available: http://arxiv.org/abs/1901.07291

[6] K. Ogueji and O. Ahia, "Pidginunmt: Unsupervised neural machine translation from west african pidgin to english," *CoRR*, vol. abs/1912.03444, 2019. [Online]. Available: http://arxiv.org/abs/1912.03444

[7] "Limburgish," https://en.wikipedia.org/wiki/Limburgish.

[8] B. Banach, "Kennis van dialect van groot belang voor niet-limburgse arts: 'een onbegrepen woord kan leiden tot een verkeerde diagnose'," *Limburger.nl*, 2021.

[9] A. Radford, K. Narashimhan, T. Salimans, and ilya Sutskever, "Improving language understanding by generative pre-training," 2018.

[10] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," 2018.

[11] P. Ramachandran, P. J. Liu, and Q. V. Le, "Unsupervised pretraining for sequence to sequence learning," 2016.

[12] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting similarities among languages for machine translation." 2013.

[13] M. Faruqui and C. Dyer, "Improving vector space word representations using multilingual correlation." 2014.

[14] C. Xing, D. Wang, C. Liu, and Y. Lin., "Normalized word embedding and orthogonal transform for bilingual word translation." 2015.

[15] S. L. Smith, D. H. Turban, S. Hamblin, and N. Y. Hammerla, "Offline bilingual word vectors, orthogonal transformations and the inverted softmax." 2017.

[16] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jegou, "Word translation without parallel data." 2018a.

[17] S. Ruder, "A survey of cross-lingual embedding models," *CoRR*, vol. abs/1706.04902, 2017. [Online]. Available: http://arxiv.org/abs/1706.04902

[18] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viegas, martin Wattenberg, and G. C. et al., "Googles multilingual neural machine translation system: Enabling zero-shot translation." 2017.

[19] M. Artetxe and H. Schwenk, "Massively multilingual sentence embeddings for zeroshot cross-lingual transfer and beyond." 2018.

[20] L. D. Guillaume Lample, Alexis Conneau and M. Ranzato., "Unsupervised machine translation using monolingual corpora only," 2018.

[21] "Wikiextractor," https://github.com/attardi/wikiextractor.

[22] Y. Kim, M. Graça, and H. Ney, "When and why is unsupervised neural machine translation useless?" *arXiv preprint arXiv:2004.10581*, 2020.

[23] L. Edman, A. Toral, and G. van Noord, "Data selection for unsupervised translation of german–upper sorbian," in *Proceedings of the Fifth Conference on Machine Translation*, 2020, pp. 1099–1103.

[24] G. Lample, M. Ott, A. Conneau, L. Denoyer, and M. Ranzato, "Phrase-based & neural unsupervised machine translation," *CoRR*, vol. abs/1804.07755, 2018. [Online]. Available: http://arxiv.org/abs/1804.07755

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[26] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *CoRR*, vol. abs/1906.08237, 2019. [Online]. Available: http://arxiv.org/abs/1906.08237

[27] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[28] [Online]. Available: $https://colab.research.google.com/github/d2l-ai/d2l-en-colab/blob/master/chapter_computational-performance/multiple-gpus.ipynb$

APPENDIX

A. Sources

Table VII shows the sources used to gather the Limburgish corpus and the parallel data.

TABLE VII: The sources used to scrape the Limburgish data. The source marked with * has been used to create the parallel data set.

Source
https://li.wikipedia.org
https://www.veldeke.net*
http://www.dewien.nl
http://www.kahuis.nl/feesnaeskes
http://www.veldeke-valkeberg.nl
http://hoebele.blogspot.nl
http://www.vlootbachtaler.nl/sjaelezeiver/wis_ger_det/wis_ger_det-nov03-juli04.html
http://www.cvdebrookhaze.nl