Highly Granular Dialect Normalization and Phonological Dialect Translation for Limburgish

Andreas Simons, Stefano De Pascale, Karlien Franco
QLVL, KU Leuven
simonsandreasjc at gmail.com, {stefano.depascale, karlien.franco}@kuleuven.be

Abstract

We study highly granular dialect normalization and phonological dialect translation on Limburgish, a non-standardized low-resource language with a wide variation in spelling conventions and phonology. We find improvements to the traditional transformer by embedding the geographic coordinates of dialects in dialect normalization tasks and use these geographically-embedded transformers to translate words between the phonologies of different dialects. These results are found to be consistent with notions in traditional Limburgish dialectology.

1 Introduction

We argue in this paper that encoding geographic coordinates as continuous parameters into transformer-based architectures allows for the improvement of normalization tasks between closely related varieties and reveals new methods in handling spatially-determined language variation.

In most tasks on multilingual data or closely related varieties, the varieties are treated on a coarse level (Dabre et al., 2020; Wu et al., 2021), without meaningfully encoding their relation to one another. The main idea behind encoding the relation between the different varieties is that knowledge transfer will take place between closely related varieties, therefore allowing for a solution to the issue of imbalanced data and a more generalized and continuous treatment of the studied varieties.

In this work we encode the geographic coordinates of approximately 1000 locations within the Limburgish language area - whose language varieties we will refer to as dialects from now on - by appending them as additional dimensions after the positional encoding in the original transformer architecture (Vaswani et al., 2023). This geographically-embedded transformer is then trained to normalize single dialect words following various different spelling conventions to a single phonetic-like spelling convention. The geographic embedding also enables the transformer to translate between any pair of Limburgish dialects on a highly granular level. We therefore separately consider the task of phonological dialect translation.

1.1 A Short Introduction to Limburgish

Limburgish is a West-Germanic language spoken by at least a million[1] native speakers in Belgium, the Netherlands and Germany. Limburgish partially underwent the High German consonant shift and has some unique features such as 3 grammatical genders, tonality, a gerund and a subjunctive in some dialects. Due to its geography and history it remained relatively isolated from both the Dutch and German standardization processes (Belemans and Keulen, 2004). Nowadays, Limburgish does not have a standard language and is superseded in official domains by the Dutch, German and French standard languages in different parts of the language area. As a result of this, Limburgish retains a complex phonology that varies continuously throughout its spoken area.

At the same time, Limburgish has been going through an atypical codification process where various standardized spelling conventions have existed since the 19th century, but often codified for individual towns. Its speakers consider all Limburgish dialects to be equally important, yet distinct varieties in what has been called a *multidialectal space* (Assendelft, 2019). This results in Limburgish texts featuring variation not only in terms of their native speakers' phonologies, but also in terms of the chosen spelling conventions. Additionally, Limburgish is one of the more extreme low-resource languages among the Germanic language family (Blaschke et al., 2023), making it very difficult to work with

[1] As per Ethnologue (2024), no elaborate estimates are known as the language only enjoys some official recognition in the Netherlands and the French-speaking community of Belgium (Limburgish Academy, 2024).

Proceedings of the Eleventh Workshop on NLP for Similar Languages, Varieties, and Dialects (VarDial 2024), pages 152–162

in most Natural Language Processing tasks.

Due to the structure of the used dataset (see Section 3), we will only consider the Limburgish dialects spoken in Belgium and the Netherlands, although there is a priori no linguistic reason to separate the dialects in Germany from the ones in Belgium and the Netherlands.[2]

2 Related Work

2.1 Limburgish NLP

NLP research on Limburgish is scarce: Nguyen and Cornips (2016) developed dialect identification for Limburgish using the Limburgish Wikipedia as a corpus. This is the only available corpus for Limburgish apart from very limited web crawl and Ubuntu localization files corpora (Blaschke et al., 2023). Michielsen-Tallman et al. (2017) is working on a Limburgish corpus which is not publicly accessible yet, and Meta's No Language Left Behind included the Maastricht dialect through its *FLORES-200* dataset (NLLB Team et al., 2022), which is now included in some applications on Hugging Face. Franco et al. (2019a,b) previously applied a statistical approach to study lexical diversity and the influence of geography on loanwords in Limburgish using the WLD (Section 3).

2.2 Dialect Normalization

Methods related to normalizing dialects using machine learning and neural approaches have been studied by Pettersson et al. (2014); Scherrer and Ljubeić (2016); Bollmann and Søgaard (2016); Honnet et al. (2018); Lusetti et al. (2018); Partanen et al. (2019); van der Goot (2021). To the best of our knowledge, no dialect normalization task has been considered where the geographic coordinates are explicitly embedded in the transformer architecture with the goal of improving knowledge transfer. Neither has such a smooth, highly granular geographic normalization task been studied. Scherrer (2011) previously studied continuous variation of Swiss-German through a statistical word generation approach. Ramponi and Casula (2023) introduced a coordinate-tagged variety corpus for Italy using Tweets and studied highly granular language identification, which was previously also considered on other languages by Han et al. (2016); Gaman et al. (2020); Chakravarthi et al. (2021).

[2]Limburgish is typically demarcated between the major Uerdinger and Benrather isoglosses within West-Germanic (Goossens, 1965). This region extends into Germany, where it is also known as Südniederfränkisch.

2.3 Dialect Translation

Character or syllable-based dialect machine translation have been considered for Swiss-German (Honnet et al., 2018), and for Japanese (Abe et al., 2018). To the best of our knowledge, no approach has considered a smooth, highly granular dialect translation task of our magnitude or considered the direct embedding of geographic coordinates for the purpose of knowledge transfer between dialects in dialect machine translation.

3 Data

The dataset used for this work is the digitized (van Hout et al., 2024) version of the *Woordenboek van de Limburgse Dialecten* (Dictionary of the Limburgish Dialects) or *WLD* (Weijnen et al., 1983-2008), an onomasiological dictionary of Limburgish, covering the dialects spoken in the Belgian and Dutch provinces of Limburg and the north of Liège. The dictionary is onomasiological in the sense that it is indexed along semantic concepts such as agrarian (e.g. ploughing, cattle), professional (e.g. bakery, mining) and general concepts (e.g. health, religion). Per semantic concept, it groups all variants per cognate, geotagged with their exact town of origin. The structure of this dictionary therefore allows us to study the phonological and orthographic variation of the Limburgish lexicon and how they interact with geography.

The WLD contains approximately 17k concepts, featuring 139k cognates and a total of 1.7M Limburgish words spread over approximately 1000 locations. About half of the entries follow a high-quality *morpho-phonological* spelling, a combination of standard Dutch orthography, the International Phonetic Alphabet (IPA) and some custom diacritics. This part of the WLD was carefully reviewed by its original curators (Weijnen et al., 1983-2008). The remainder follows various spelling conventions from local dictionaries or standardized conventions such as the Veldeke spelling (Bakkes et al., 2003).

The locations are tagged with *kloeke* codes, geographic tags that refer to all locations in Belgium, the Netherlands, northern France and western Germany. We converted these kloeke codes to geographic coordinates, which were then normalized to unit intervals. Entries corresponding to locations that were clearly outside the Limburgish area were omitted. We carried out some preprocessing and cleaning steps on the data such as deleting entries

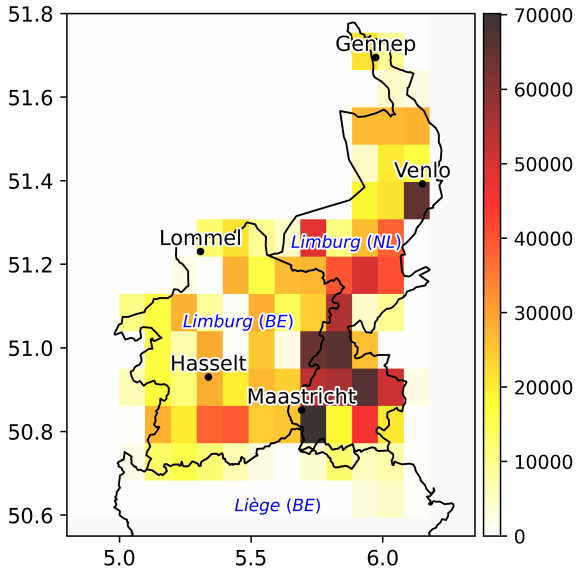

Figure 1: 2D frequency histogram of the geographic spread of entries in the WLD.

with overly noisy characters (as a result of poor digitization), splitting sentence entries into individual words and omitting superfluous characters such as punctuation marks. The geographic spread of the resulting data are shown in Fig 1. Finally, we applied some manually curated rules to resolve predictable digitization errors, such as converting incorrect ASCII characters.

3.1 Task-Specific Datasets

Since we did not have access to curated datasets or parallel corpus data for Limburgish or any of its dialects, two new datasets needed to be generated from the WLD for the normalization and phonological dialect translations tasks.

For the **normalization task**, we aimed to train a model that correctly converts the characters of any of the Limburgish (conventional) spelling systems to the high quality morpho-phonological standards that constitute approximately half the WLD. These standards closely resemble IPA, and enable further study of Limburgish text data, which is often blurred by localized spelling conventions.

We therefore split the dataset into words following an accurate phonetic spelling convention and words with local or other conventional spelling conventions. This is achieved using manually selected filters containing typical conventions in the Veldeke spelling and other local conventions that are not known within the WLD's phonetic system, such as the use of *ieë*, *aa* or *ää*. Any words that do not contain any n-grams which are exclusively used in conventional spelling are then assumed to be in phonetic notation, which a manual inspection confirms. This results in an approximately equal split in normalized-unnormalized data. For each unnormalized entry, cognates of nearby dialects are then selected as the normalized equivalents. By defining a nearby dialect as being within a 0.5 km radius, we ensure that the variation is more likely due to spelling conventions rather than a change in phonology between the two dialects. Typically, the phonologies of Limburgish dialects stay consistent within such a radius, unless major isoglosses are crossed. This results in a dataset of 118k unnormalized-normalized pairs, with an average pair distance of 0.39 km. An example of such a pair is given below (with both words originating from the dialect of Echt).

> *kroedwès- krutweš* (without translation; a folkloristic herb)

For the task of **phonological dialect machine translation**, we again chose a character-based approach and trained a model to translate the phonology between any dialects, as this is the largest source of variation in Limburgish apart from spelling variation. We only used the part of the WLD dataset that is already normalized (approximately 805k entries), therefore avoiding arbitrary spelling conventions, and generated a new dataset. We paired each word in this normalized subset with all cognates from other dialects in the same dataset. Very frequent words such as *in* (English: in) or *van* (English: of) were omitted, as were words that are rare in other dialects (a frequency of <10). For each family of cognates, we undersampled the available cognates due to the imbalance in geographic representativeness of the data (see Fig 1). The undersampling was done by weighting the geographic frequency distribution of the cognates according to a 2D Gaussian kernel smoothing with a bandwidth factor of 0.2 and then undersampling by 70%.

As the entries in the new dataset grow quadratically with the number of cognates, a 10% subset is sampled of all pairs in the new dataset, resulting in a phonological dialect translation dataset of 20.2M entries. For example, the word *šo.l* (originating from Bree) is paired with 85 different cognates from other dialects:

> *sxo:l* (Grote-Brogel), *šo̜l* (Kanne), *šuəl* (Kerkrade), *šo.əl* (Valkenburg), *sxo.l* (Nederweert) ...

4 Methods

4.1 Encoding

We tried two encoding methods: a simple one-hot encoding and an experimental method using phonological vectors from the PanPhon library (Mortensen et al., 2016) in Python. The main idea was that encoding 24 articulatory features is more meaningful and compact for data that varies greatly phonologically. The phonological encoding was surprisingly outperformed by a simple one-hot encoding in all experiments. This was likely due to the high complexity of the phonology of some dialects. For example, the dialect of Weert has 28 vowels over 5 heights, for which PanPhon's binary vowel height system is insufficient.

We opted for a 92-dimensional one-hot encoding, corresponding to all unique characters that remained after the preprocessing and cleaning steps. Due to the complexity of Limburgish phonology, many special diacritics are featured to realize the correct vowels or tonality. These diacritics are represented as separate characters. All words above 10 characters were omitted, and shorter words were padded to 10-dimensional vectors.

4.2 Coordinate Embedding in Transformer

The modification to the traditional transformer architecture was done in Tensorflow's functional API v2.12 (Abadi et al., 2015), Keras v2.15 (Chollet and et al., 2015), and KerasNLP v0.4.1 (Watson et al., 2022). Typically when discrete language tokens are used for multilingual models (and therefore different from our approach), this is done at the tokenizer level, thereby increasing the input vocabulary dimension or dimension of the input embedding. We instead pass the geographic coordinates as two extra dimensions directly after the positional encoding, resulting in a similar number of weights and training complexity when compared to the traditional transformer architecture. The (rescaled) coordinates are only appended to the first dimension of the embedded vector (after the input embedding and positional encoding) to keep the data sparse and can be visualized as

$$\begin{bmatrix} e_{1,1} & e_{1,2} & \dots & e_{1,9} & e_{1,10} \\ \vdots & \vdots & & \vdots & \vdots \\ e_{N,1} & e_{N,2} & \dots & e_{N,9} & e_{N,10} \\ y & 0 & \dots & 0 & 0 \\ x & 0 & \dots & 0 & 0 \end{bmatrix},$$

where $e_{i,j}$ represent the floats of the embedded vector after the input embedding and positional encoding, N the dimension of the embedding vector space and x, y the rescaled geographic coordinates.

For the encoder block, the coordinates corresponding to the input word are embedded. For the decoder block, the first input (using autoregression) is the embedded start token with the coordinates of the target word appended as two extra dimensions.

4.3 Evaluation

We consider two evaluation metrics::

Levenshtein ratio: the Levenshtein ratio between two words s_1 (reference) and s_2 (hypothesis) is defined as (Bachmann, 2021)

$$1 - \frac{\text{Levenshtein distance}(s_1, s_2)}{\text{len}(s_1) + \text{len}(s_2)} \quad (1)$$

where the Levenshtein distance between s_1 and s_2 is defined as the number of single-character insertions, deletions, and substitutions required to transform s_1 into s_2. The Levenshtein ratio is a character-based measure of similarity between two words, normalized for the lengths of the words (unlike the typical Levenshtein distance). Two identical words have a Levenshtein ratio of 1, the minimum ratio is 0.

CharacterF: *character n-gram F-score* or *chrF* (Popović, 2015) is the character-based machine translation equivalent of the traditional F-score. As it relies on character n-grams, it is more sensitive towards morpho-syntactic phenomena. The chrF score between two words s_1 (reference) and s_2 (hypothesis) is defined as (Popović, 2015)

$$\text{chrF} = 2\frac{\text{chrP} \cdot \text{chrR}}{\text{chrP} + \text{chrR}} \quad (2)$$

where chrP is the percentage of n-grams from s_2 that can be found in s_1 and chrR the percentage of n-grams from s_1 in s_2. We use 3-grams as these correspond closely to human judgment (Popović, 2015).

We initially expect both these metrics to undervalue the performance in this task; the WLD is very rich in diacritics and it is undesirable that the predicted normalized word is penalized for using a diacritic that is phonologically very close, but not identical to the expected diacritic. To mitigate this, we will also compute these metrics after stripping the diacritics, and thus only considering the ASCII characters. An example of how these metrics behave can be found in Table 1.

Unnormalized - Normalized (translation)	ChrF	Levenshtein ratio	ChrF (no diacritics)	Levenshtein ratio (no diacritics)
vief - viêf (five)	0	0.75	1	1
kroedwès - krutweš (herb)	0	0.4	0.18	0.53
waere - wĕre (to become)	0	0.8	0.4	0.8
aafdoeë - āfdūə (to mow grass)	0	0.33	0.25	0.33
schoppen - sxopə (to kick)	0	0.46	0	0.46

Table 1: A sample of the normalization dataset and their evaluation according to the selected metrics.

We observe that chrF is generally much too strict, while chrF with diacritics removed is significantly more tolerant. The Levenshtein ratio seems more tolerant than the non-diacritic chrF, while the non-diacritic Levenshtein ratio seems the most tolerant metric. A more in-depth manual analysis showed that the non-diacritic chrF metric corresponded closest to human judgement.

An inherent difficulty of working with Limburgish data is discerning the variation caused by differences in phonology from the variation caused by different spelling conventions, which is also a barrier for any other language that varies phonologically and orthographically (usually the case for non-standardized languages or families of dialects). To establish some baselines, we determined a lower boundary for all four metrics by measuring them on the unnormalized-normalized word pairs in the dataset, reflecting the accuracy when the same input were to be predicted. We also determined an upper boundary by estimating the inherent variation in spelling conventions: we computed the four metrics for all cognates within a radius of 6 km of each unnormalized word in the dataset. The assumption is that most remaining variation will then be due to differences in orthography, rather than phonology. This baseline therefore indicates the maximally attainable values for these metrics. We found the following lower and upper boundaries:

	ChrF	ChrF no diac.	Lev.	Lev. no diac.
Lower	0.112	0.242	0.599	0.710
Upper	0.440	0.589	0.751	0.84

Table 2: Expected lower and upper boundaries for the evaluation metrics.

Due to a lack of any curated data for Limburgish and the inherent variation in the data, these boundaries and a manual analysis in Section 6 are our best available approaches for evaluation, for a more elaborate discussion we refer to Section 8.

4.4 Normalization Task

To test whether embedding geographic coordinates improves the traditional transformer architecture, we first ran a hyperparameter search on the task using a traditional transformer without coordinate embedding. Using the traditional transformer for this task is possible since the target words follow the (relatively) uniform morpho-phonological spelling of the WLD and no dialect or spelling variation is required from the decoding part. We split the data in a $80-10-10$ train, validation, test dataset and varied stacking of encoder and decoder blocks from $1-5$, the embedding dimension from $1-1024$, the latent dimension from $1-1024$ and the number of attentions heads in each block from $1-16$ using the Optuna library (Akiba et al., 2019). We used the Adam training method and a Sparse Categorical Crossentropy metric and ran 100 iterations using Optuna's *Tree Parzen Estimator*. The optimized set of parameters was then used to train the traditional transformer and the geographically-embedded transformer and compare their performance on the test set.

The optimized traditional transformer has a total of 5.1M parameters, the geographically-embedded transformer has 5.2M parameters due to the extra 2 dimensions after the positional encoding step. These additional parameters only allow for a heterogeneous interaction between the coordinates and the embedded characters in the attention mechanism, and do not allow for any further inference of information in the attention mechanism that could otherwise be associated with having slightly more parameters.

4.5 Phonological Dialect Translation Task

Unlike the previous task, performance in the phonological dialect translation cannot be readily compared to the traditional transformer architecture as it does not natively allow for variation of the

target dialect. We therefore only considered the geographically-embedded transformer.

We again used an $80-10-10$ train, validation, test dataset split but did not run a hyperparameter search due to resource constraints. We instead used standard parameter values such as an embedding dimension of 256, a latent dimension of 1024, 8 attention heads, and no stacked encoder or decoder blocks, resulting in a total of 7.6M parameters. For the optimizer and loss we again opted for Adam with a Sparse Categorical Crossentropy loss.

5 Results

5.1 Normalization Task

The hyperparameter search for the traditional transformer architecture yielded the following parameters: 2 layers of stacked encoder/decoder blocks, an embedding dimension of 150, a latent dimension of 1000, and 7 attention heads, resulting in a total of 5.1M parameters. The evaluation metrics on the test sets for the traditional transformer and the architecture with geographic coordinates embedded can be found in Table 3. We also present some representative examples of the geographically-embedded transformer's performance when normalizing the test set, along with the non-diacritic chrF metric in Table 5.

coords	ChrF	ChrF no diac.	Lev.	Lev. no diac.
no	0.353	0.506	0.713	0.817
yes	**0.363**	**0.516**	**0.718**	**0.821**

Table 3: The evaluation metrics on the test set for the traditional transformer and the transformer with geographic coordinates embedded.

5.2 Phonological Dialect Translation Task

The evaluation of the geographically-embedded transformer on the phonological dialect translation task can be found in Table 4. We again present some representative examples of the translation task with their corresponding non-diacritic chrF metrics and the locations of the input and target dialects in Table 6.

ChrF	ChrF no diac.	Lev.	Lev. no diac.
0.407	0.485	0.687	0.736

Table 4: The evaluation metrics on the test set for the phonological translation task.

6 Discussion

6.1 Normalization Task

As can be seen in Table 3, the geographically-embedded transformer outperforms the normal transformer according to all metrics that we measured. The results are statistically significant ($p < 0.001$) according to two-sided Wilcoxon hypothesis tests. The improvements to the traditional transformer's performance are most prominent in ascending order of 'tolerance' of the metrics, as we could have expected. When comparing these results with our established lower and upper boundaries (Table 2), we again find that the upper boundaries are approached more closely by the more tolerant metrics. A geographic analysis of the evaluation metrics showed that there is no geographic bias, as the performance is relatively homogeneously spread.

Manually analyzing a sample of the geographically-embedded transformer's predictions (Table 5), we find that the model generally succeeds in correctly normalizing various Limburgish spelling conventions to a phonetic spelling. For example, in entry 3 (*daavekot* → *dāvəkọt*), the long *aa* is normalized to *ā*, the *e* to the schwa and the *o* to the correct Limburgish phoneme.

The model also abides by well-known notions in Limburgish dialectology: in entry 1 (*sjnaps* → *snaps*), the *sj* is normalized to an *s*, even though this is a neologism derived from High German, showing that the model correctly applies the Panninger isogloss within Limburgish that is associated with the *s*→*ʃ* rule (Bakkes et al., 2007). In entry 7 (*kool* → *kiəl*), the unnormalized word uses the Dutch phoneme *o* which does not occur for that word in Limburgish, but the model correctly predicts *iə*.

In other instances such as as 5 and 13, the model predicts normalized words that are more accurate than the original target normalizations. This is due to the fact that we generated this dataset ourselves without a very elaborate manual curation, as we did not have access to a curated or golden standard dataset. Despite inaccuracies in the generated dataset, the model has generalized well to avoid conventional spelling: in entry 3 an *ò* is included in the target, which is not part of the phonetic notation used in the WLD. The model instead correctly normalized this phoneme to ọ. The evaluation met-

	Unnormalized word	Prediction	Target	ChrF (no diac.)	Translation
1	*sjnaps*	*snaps*	*snàps*	1.0	schnaps (drink)
2	*zeik*	*zɛi̯.k*	*zɛi̯.k*	1.0	fecal sludge
3	*daavekot*	*dāvəkọt*	*dāvəkòt*	1.0	dovecote
4	*sjollek*	*šolək*	*šọlək*	1.0	type of apron
5	*volle*	*vọlə*	*vọl*	0.667	full
6	*strooie*	*stroi̯ə*	*strōən*	0.5	of straw (material)
7	*kool*	*kiəl*	*kīəl*	1.0	cabbage
8	*hèndichə*	*hendix*	*hendixe*	0.889	handy
9	*kwartsche*	*kwartse*	*kwē<rtšə*	0.2	quarter
10	*hiemël*	*hi:məl*	*hi:məl*	1.0	heaven
11	*lintteeke*	*lintēkə*	*lentēkə*	0.6	scar
12	*sjei*	*šɛi̯*	*šɛi̯*	1.0	vagina (horse)
13	*tweede*	*twēdə*	*de*	0	second
14	*preuvə*	*prèùve*	*prēūvə*	0.75	to taste
15	*áfzéttə*	*ifzetə*	*afzɛtə*	0.75	to rip off/defraud

Table 5: A random sample of the geographically-embedded transformer's performance on the test set.

	Input	Prediction	Target	ChrF (no diac.)	Translation	Locations
1	*špat*	*spat*	*spat*	1.0	osteoarthritis (horse)	Moresnet→Meeuwen
2	*rempəl*	*rimpels*	*rumpels*	0.6	wrinkles	Meterik→Blerick
3	*moder*	*mojər*	*mojər*	1.0	mother	Bocholt→Millen
4	*xeld*	*xɛ̄lt*	*xēld*	0.5	money	Gulpen→Moelingen
5	*bọtərham*	*botəram*	*botəram*	1.0	sandwich	Achel→Blitterswijck
6	*werk*	*werk*	*werk*	1.0	work	Sittard→Beverst
7	*bɛ̣səl*	*bɛ̣səl*	*bɛ̣.səl*	0.286	bushel (hay)	Beverst→Munsterbilzen
8	*wɛ̣x*	*wex*	*wex*	1.0	road	Landen→Venray
9	*hō̩tə*	*họwtə*	*họwtən*	0.857	wooden	Genk→Neeroeteren
10	*briə.kə*	*brɛ̄.kə*	*brɛ̄.kə*	1.0	to spread manure	Jesseren→Nerem
11	*kát*	*kat*	*kat*	1.0	cat	Wijchmaal→Blitterswijck
12	*slɛi̯p*	*slɛi̯.p*	*slɛi̯.p*	1.0	field drag	Neeritter→Bocholt
13	*wilde*	*wel*	*wøl*	0	wild	Ophoven→Lummen
14	*nak*	*nek*	*nek*	1.0	neck	Munsterbilzen→Horst
15	*hūs*	*hōēs*	*hôêês*	0.4	house	Gruitrode→Lottum

Table 6: A random sample of the geographically-embedded transformer's performance on the phonological dialect translation test set.

rics penalize these instances, even though they are desirable for our purposes.

The model fails in a few instances in predicting the correct normalization: in 8 the wrong gender/plural is predicted, in 9 (pred.: *kwartse*, target: *kwē<rtšə*) and in 15 (pred.: *ifzetə* target: *afzɛtə*) some characters are incorrectly normalized. In entry 14, conventional spelling is used in the prediction (*prèùvə*).

From this manual analysis and the close correspondence to the estimated upper boundaries we can conclude that the geographically-embedded architecture is appropriate for normalization of Limburgish spelling to phonetic notation and an improvement over the traditional transformer architecture for this purpose.

6.2 Phonological Dialect Translation Task

The evaluation metrics (Table 4) approach the estimated upper boundaries, but not as closely as in the normalization task. There is again no geographic bias as the evaluation metrics are homogeneously

spread over the studied language area. A manual analysis of a sample of some predictions (Table 6) shows that the model succeeds in correctly translating the phonology of various words from one Limburgish dialect to another: in for example entry 1 (*špat* → *spat*), the *s*→∫ rule associated with the Panninger isogloss is correctly applied. In entry 10 (*briə.kə* → *brɛ̄ᵩ.kə*), the correct sound *ɛ̄ᵩ* is translated for Nerem, even though the place of origin Jesseren is only 10 km removed and uses the sound *iə*.

In most instances, the translation matches the expected target. In some entries such as 2 and 13 the model corrects the input: entry 2 is an incorrect singular as the other cognates are plurals and the missing pluralization is therefore an error in the dataset, entry 13 is likewise a transcription error in the WLD and should have been *wel*.

In entries such as 4, 9, and 15 the model also performs better than the dataset and correctly predicts phonetic notation: *t* instead of *d* for entry 4 (pred.: *xɛ̄ᵩlt* target: *xēld*), no end-*n* for entry 9 (pred.: *hǫwtə* target: *hǫwtən*) and no conventional Limburgish spelling in the case of entry 15 (pred.: *hōēs* target: *hôêês*).

In some instances the model fails to predict the correct diacritics, such as in entry 7 (pred.: *bɛᵩsəl* target: *bɛᵩ.səl*), but most of the low evaluation metrics correspond to instances where the model predicted more desirable results than the dataset provided as target words.

From this manual analysis and the evaluation metrics we can conclude that phonological dialect translation was successfully achieved using the geographically-embedded transformer.

6.3 Language Variation Maps

The phonological translation model allows us to specify any input and target coordinates for the translation task; by fixing an input word and coordinates and varying the target coordinates we can generate highly granular language variation maps that show the phonological variation of the input word/phoneme. These maps can be used to extensively compare the model's predictions to traditional dialectological maps, or to study generalizations the model has learned due to the inherent language variation. In Fig 2 we provide such a map with all variation of *school* for fixed location Bree with input word *šo.l*. Overlaid on this map we indicate the natural variation of Limburgish phonology using dots, sourced from the WLD. We notice the close correspondence to the natural phonological

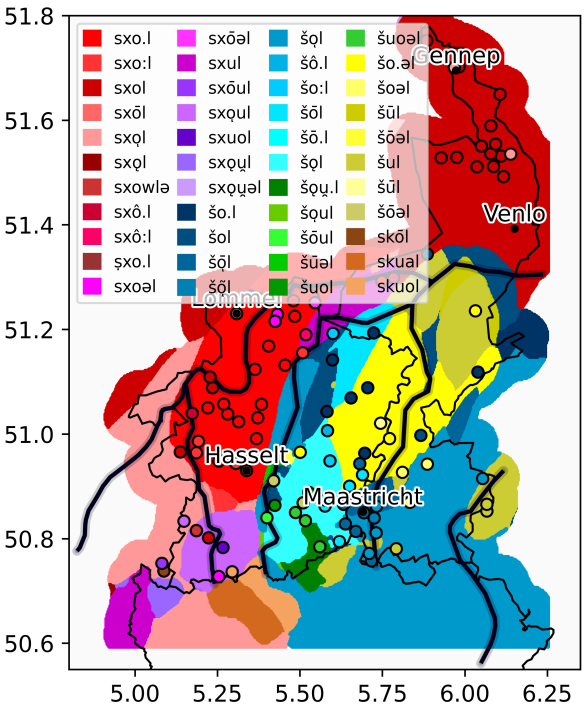

Figure 2: Generated variation map for *school* (*šo.l*) with Bree as fixed input. The overlaid dots correspond to the natural variation as found in the WLD. The color scheme purposely groups closely related phonemes. The black lines indicate the major isoglosses in the Limburgish area (Bakkes et al., 2007).

variation, in particular the clear separation between variants starting with *sx* and *š*. This isophone is part of the Panninger side line, an important isogloss in the Limburgish language area (in black, right next to Hasselt and converging in the north with the other isoglosses).

6.4 Availability

The trained models, datasets, figures and GIS data will be made available on `github.com/AndreasJCSimons/LimburgishNLP`.

7 Conclusion

We found that embedding geographic coordinates after the positional encoding allows us to normalize highly phonologically and orthographically varying data more accurately than the traditional transformer architecture. Additionally, we found that this geographic embedding allows us to translate the phonology of words between any Limburgish dialects and to generate language variation maps that can be compared to traditional dialectology or to study generalized phonological patterns that the model has implicitly learned.

8 Limitations

This work is limited by the lack of properly curated datasets and methodologies to evaluate the performance of dialect normalization and translation tasks, which hinders a more accurate evaluation of the used methods. We therefore had to evaluate the trained models using a manual analysis and estimates for an expected upper boundary on some evaluation metrics, given the inherent phonological and orthographic variation in the data. In Subsections 6.1 and 6.2 it is clear that in some instances the dataset is of low quality. However, due to the size of the dataset it is likely that the model has generalized beyond the low-quality entries: this can be seen both in the manual analysis where the model corrects wrong targets (even though it is penalized by the loss function) as well as in the language variation maps of Subsection 6.3 where the model has correctly learned Limburgish sound changes.

Another limitation is that the normalization and phonological dialect translation tasks only took the spelling and phonology of the words into account and not their semantics. While this rarely resulted in inaccurate predictions, a more elaborate normalization or translation scheme should take semantic information into account, as this can sometimes be tied to phonological patterns. For example, High German loanwords such as *sjnaps* (Table 5) are typically not subject to internal Limburgish sound changes and remain invariant.

Finally, the data of the WLD is not fully synchronous: it contains older dialect surveys such as the data from the Willems survey (19th century), SGV (1914), and ZND (from 1922 onwards). Additionally, data was collected in Belgian Limburg from the 1960s onwards to match missing data with respect to Dutch Limburg (Weijnen et al., 1983-2008). This means that the data collection occurred during a period of a major linguistic shift: between 1950 and 1980 a period of hyperstandardization occurred in Belgium that sought to promote *Algemeen Beschaafd Nederlands* (General Civilized Dutch) and stigmatize any other languages or language variation (Hoof and Jaspers, 2012). We also did not have access to any data from beyond the Dutch-German border, even though there is linguistically no reason to separate the dialects spoken between the Uerdinger and Benrather lines in Germany from the dialects in Belgium and the Netherlands.

9 Ethics Statement

This work complies with the ACM Code of Ethics and Professional Conduct (https://www.acm.org/code-of-ethics) with particular attention to articles 1.1 and 1.4: many underserved languages and language communities exist, and language variation and diversity is itself an exercise in low-resource NLP. By contributing to the research of non-standardized languages, low-resource languages or methods in NLP that can handle language variation, we hope to provide instruments that may be beneficial to disadvantaged languages communities.

The data used in this work, the *Woordenboek van de Limburgse Dialecten*, was manually processed over many years using dialect surveys and native speakers, who have been anonymized in the final dataset. Regardless, we are aware that by the very nature of this research, i.e. highly granular geographic analysis of language variation using methods in Deep Learning, we are studying phenomena that are tied to a person's native dialect, upbringing and socioeconomic situation. It is worrying that in recent years this has been abused for purposes of surveillance. For example, language variation has been used in dialect identification software by countries to evade privacy regulations during asylum procedures (European Digital Rights et al., 2021).

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org.

Kaori Abe, Yuichiroh Matsubayashi, Naoaki Okazaki, and Kentaro Inui. 2018. Multi-dialect neural machine translation and dialectometry. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework.

Brenda Assendelft. 2019. De codificatie van het Limburgs. *Taal en Tongval*, 71(1):1–30.

Max Bachmann. 2021. Levenshtein module. Accessed: 10/03/2024.

Pierre Bakkes, Rob Belemans, Georg Cornelissen, Ronny Keulen, Ton Van de Wijngaard, Herman Crompvoets, and Frans Walraven. 2007. Riek van klank: inleiding in de Limburgse dialecten.

Pierre Bakkes, Herman Crompvoets, Jan Notten, and Frans Walraven. 2003. Spelling 2003 voor de Limburgse dialecten. Accessed: 10/03/2024.

Rob Belemans and Ronny Keulen. 2004. *Belgisch-Limburgs*. Taal in stad en land. Lannoo.

Verena Blaschke, Hinrich Schuetze, and Barbara Plank. 2023. A survey of corpora for Germanic low-resource languages and dialects. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 392–414, Tórshavn, Faroe Islands. University of Tartu Library.

Marcel Bollmann and Anders Søgaard. 2016. Improving historical spelling normalization with bi-directional LSTMs and multi-task learning. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 131–139, Osaka, Japan. The COLING 2016 Organizing Committee.

Bharathi Raja Chakravarthi, Gaman Mihaela, Radu Tudor Ionescu, Heidi Jauhiainen, Tommi Jauhiainen, Krister Lindén, Nikola Ljubešić, Niko Partanen, Ruba Priyadharshini, Christoph Purschke, Eswari Rajagopal, Yves Scherrer, and Marcos Zampieri. 2021. Findings of the VarDial evaluation campaign 2021. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 1–11, Kiyv, Ukraine. Association for Computational Linguistics.

François Chollet and et al. 2015. Keras. `https://keras.io`.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A Comprehensive Survey of Multilingual Neural Machine Translation.

Ethnologue. 2024. Limburgish. Accessed: 10/03/2024.

European Digital Rights, Access Now, Migration, Technology Monitor, Platform for International Cooperation on Undocumented Migrants, and Statewatch. 2021. Uses of AI in migration and border control: A fundamental rights approach to the Artificial Intelligence Act. Accessed: 10/03/2024.

Karlien Franco, Dirk Geeraerts, Dirk Speelman, and Roeland van Hout. 2019a. Concept characteristics and variation in lexical diversity in two Dutch dialect areas. *Cognitive Linguistics*, 30(1):205–242.

Karlien Franco, Dirk Geeraerts, Dirk Speelman, and Roeland van Hout. 2019b. Maps, meanings and loanwords: The interaction of geography and semantics in lexical borrowing. *Journal of Linguistic Geography*, 7(1):14–32.

Mihaela Gaman, Dirk Hovy, Radu Tudor Ionescu, Heidi Jauhiainen, Tommi Jauhiainen, Krister Lindén, Nikola Ljubešić, Niko Partanen, Christoph Purschke, Yves Scherrer, and Marcos Zampieri. 2020. A report on the VarDial evaluation campaign 2020. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 1–14, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

Jan Goossens. 1965. Die Gliederung des Südniederfränkischen. In *Rheinische Vierteljahrsblätte*, pages 79–94.

Bo Han, Afshin Rahimi, Leon Derczynski, and Timothy Baldwin. 2016. Twitter geolocation prediction shared task of the 2016 workshop on noisy user-generated text. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 213–217, Osaka, Japan. The COLING 2016 Organizing Committee.

Pierre-Edouard Honnet, Andrei Popescu-Belis, Claudiu Musat, and Michael Baeriswyl. 2018. Machine translation of low-resource spoken dialects: Strategies for normalizing Swiss German. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Sarah Van Hoof and Jürgen Jaspers. 2012. Hyperstandaardisering. *Tijdschrift voor Nederlandse Taal- en Letterkunde*, 128(1):97–125.

Limburgish Academy. 2024. Limburgish Language. Accessed: 10/03/2023.

Massimo Lusetti, Tatyana Ruzsics, Anne Göhring, Tanja Samardžić, and Elisabeth Stark. 2018. Encoder-decoder methods for text normalization. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 18–28, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yuri Michielsen-Tallman, Ligeia Lugli, and Michael Schuler. 2017. A Limburgish Corpus Dictionary: Digital Solutions for the Lexicography of a Non-standardized Regional Language.

David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori Levin. 2016. PanPhon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484, Osaka, Japan. The COLING 2016 Organizing Committee.

Dong Nguyen and Leonie Cornips. 2016. Automatic detection of intra-word code-switching. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 82–86, Berlin, Germany. Association for Computational Linguistics.

NLLB Team et al. 2022. No Language Left Behind: Scaling Human-Centered Machine Translation.

Niko Partanen, Mika Hämäläinen, and Khalid Alnajjar. 2019. Dialect text normalization to normative standard Finnish. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 141–146, Hong Kong, China. Association for Computational Linguistics.

Eva Pettersson, Beáta Megyesi, and Joakim Nivre. 2014. A multilingual evaluation of three spelling normalisation methods for historical text. In *Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 32–41, Gothenburg, Sweden. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Alan Ramponi and Camilla Casula. 2023. DiatopIt: A corpus of social media posts for the study of diatopic language variation in Italy. In *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pages 187–199, Dubrovnik, Croatia. Association for Computational Linguistics.

Yves Scherrer. 2011. Morphology Generation for Swiss German Dialects. In *Systems and Frameworks for Computational Morphology*, pages 130–140, Berlin, Heidelberg. Springer Berlin Heidelberg.

Yves Scherrer and Nikola Ljubeić. 2016. Automatic normalisation of the Swiss German ArchiMob corpus using character-level machine translation. In *Conference on Natural Language Processing*.

Rob van der Goot. 2021. CL-MoNoise: Cross-lingual lexical normalization. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 510–514, Online. Association for Computational Linguistics.

Roeland van Hout, Nicoline van der Sijs, Erwin Komen, Henk van den Heuvel, and et al. 2024. Elektronisch Woordenboek van de Limburgse Dialecten (e-WLD). Accessed: 10/03/2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Matthew Watson, Chen Qian, Jonathan Bischof, François Chollet, and et al. 2022. Kerasnlp. `https://github.com/keras-team/keras-nlp`.

Antonius Weijnen, Jan Goossens, Pieter Goossens, Joep Kruijsen, Har Brok, Jo Kokkelmans, Herman Crompvoets, Jan van Schijndel, Jos Molemans, Joke Verbeek, Miet Ooms, Ton van de Wijngaard, J. Busch, Ronny Keulen, and Mariëtte Lubbers. 1983-2008. *Woordenboek van de Limburgse dialecten*. Van Gorcum/Gopher Publishers, Assen/Amsterdam/Maastricht/Utrecht.

Liwei Wu, Shanbo Cheng, Mingxuan Wang, and Lei Li. 2021. Language tags matter for zero-shot neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3001–3007, Online. Association for Computational Linguistics.