HiTZ at VarDial 2025 NorSID: Overcoming Data Scarcity with Language Transfer and Automatic Data Annotation

Jaione Bengoetxea Mikel Zubillaga Ekhi Azurmendi
Maite Heredia Julen Etxaniz Markel Ferro Jeremy Barnes
HiTZ Center – Ixa, University of the Basque Country (UPV/EHU)
name.surname@ehu.eus

Abstract

In this paper we present our submission for the NorSID Shared Task as part of the 2025 VarDial Workshop (Scherrer et al., 2025), consisting of three tasks: Intent Detection, Slot Filling and Dialect Identification, evaluated using data in different dialects of the Norwegian language. For Intent Detection and Slot Filling, we have fine-tuned a multitask model in a cross-lingual setting, to leverage the xSID dataset available in 17 languages. In the case of Dialect Identification, our final submission consists of a model fine-tuned on the provided development set, which has obtained the highest scores within our experiments. Our final results on the test set show that our models do not drop in performance compared to the development set, likely due to the domain-specificity of the dataset and the similar distribution of both subsets. Finally, we also report an in-depth analysis of the provided datasets and their artifacts, as well as other sets of experiments that have been carried out but did not yield the best results. Additionally, we present an analysis on the reasons why some methods have been more successful than others; mainly the impact of the combination of languages and domain-specificity of the training data on the results.

1 Introduction

Dialectal variation is ubiquitous in human language and should be taken into account when performing Natural Language Processing (NLP) tasks, as NLP systems unable to deal with dialectal data can cause users to feel frustrated and lead to unintended biases (Harwell, 2018).

This is especially relevant for Spoken Language Understanding (SLU), a field of Speech Processing and Natural Language Understanding aimed at ensuring the semantic comprehension of human utterances by virtual assistants. To make systems that rely on SLU more robust and able to handle real use-cases, it is necessary to develop resources for these tasks not only for different languages, but for different language varieties, so that the benefits of these models can reach a wider variety of speech communities.

With this motivation, the NorSID Shared Task consists of three subtasks (intent detection, slot filling and dialect identification) in four Norwegian variants: Bokmål (B), Western (V), Trøndersk (T) and North Norwegian (N). The tasks are centered around common virtual assistant tasks, such as setting alarms or questions about the weather.

Our team participated in all three subtasks, for a total of 6 runs: 3 for the SID (Slot and Intent Detection) tasks and 3 for Dialect Identification. As a team, we placed first in Dialect Identification, second in Intent Detection, and third in Slot Filling. Our code is publicly available on GitHub.[1]

2 Task Descriptions

As mentioned, this shared task consists of the following three subtasks:

Intent Detection. It is a text classification task that assigns intent labels to the utterances of the users, to guide the chatbot's answer, depending on its domain and purpose.

Slot Filling. It requires classifying token spans that contain relevant information for a virtual assistant to fulfill certain tasks, e.g., to set an alarm, the assistant needs to know the time to set it to.

Dialect Identification. The aim of this classification task is to identify the dialect of the utterance.

2.1 Initial Data: NoMusic Dataset

The shared task uses the NoMusic dataset (Mæhlum and Scherrer, 2024), a "multi-parallel resource for written Norwegian dialects, and the first evaluation dataset for slot and intent detection focusing on non-standard Norwegian varieties."

[1] https://github.com/hitz-zentroa/vardial-2025

Proceedings of the Twelfth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 209–219
January 19, 2025.

id	text	intent	dialect	slots
90/9	Sett alarm for kl. 6	alarm/set_alarm	V	datetime
45/2	Skal d bli sol i dag?	weather/find	N	weather/attribute datetime
183/2	Æ vil gje boka 3 stjenre .	RateBook	N	object_type rating_value rating_unit

Table 1: Random examples from the NoMusic development set.

To construct the development and test set (3300/5500 instances each), 11 Norwegian translators manually translated phrases from the corresponding English xSID sets (van der Goot et al., 2021) into four different Norwegian dialects (North Norwegian, Trøndersk, West Norwegian and Bokmål). Shared task participants only had access to the dev set during the competition. See Table 1 for examples from the dev set and Table 2 for the distribution of labels.

For training data, a machine-translated version of the English xSID train set (43,605 instances) was provided.[2] The instances have been translated into Bokmål and are annotated for both the intent detection and slot filling tasks. It preserves the original intent labels and the slots have been projected from one language to the other, although the shared task organizers report that the quality of both the translation and the annotation projection is relatively poor.

Dialect	Dev	Test	Dist
West Norwegian (V)	1,500	2,500	45.45%
Trøndersk (T)	900	1,500	27.27%
North Norwegian (N)	600	1,000	18.18%
Bokmål (B)	300	500	9.09%
Total	3300	5500	100%

Table 2: Distribution of dialect tags in the NorSID development and test sets. Notice that the data distribution is highly skewed towards West Norwegian.

3 Intent Detection & Slot Filling

In this section, we will detail our participation in the intent and slot filling subtasks. We first explain the data (Section 3.1) and the experimental design (Section 3.2), and finally a description and an analysis of our results (Section 3.3).

[2]More details of xSID are presented in Section 3.1.

3.1 Data

xSID (van der Goot et al., 2021; Aepli et al., 2023; Winkler et al., 2024) is a cross-lingual corpus for SLU.[3] The original English data was sampled by selecting random instances from the Snips (Coucke et al., 2018) and Facebook (Schuster et al., 2019) datasets. It features annotations for both intent detection, with one intent per instance; and slot filling, using the BIO format to tag each token. For the validation and test sets, the data was manually translated by native speakers of each language, maintaining the original intents, while the slots were manually re-annotated. The training data is available for most of the xSID languages through machine translation and projection of the slots.

For the Intent Detection task, there are a total of 18 intents. As per the slot filling task, there are 33 possible slots that can appear as the beginning (B) or inside (I) of a span and an O tag for the absence of entity. This results in a total of 67 possible tags.

Although the original paper leaves duplicated sentences to model the natural distribution found in the data, we deduplicate to avoid our models overfitting on the training data. We only carry out shallow deduplication, removing instances that contain the same text.

3.2 Experiments

Intent detection and slot filling are two highly related tasks. In fact, there are some slots that will only appear in sentences tagged with a certain intent and vice-versa. In this respect, a model could make use of the annotations of both tasks at the same time to obtain better predictions. Our experiments for the SID tasks build on that idea, using a multilingual multitask model jointly trained for intent detection and slot filling. As shown in Figure 1, our multitask models learn to classify the intents on top of the [CLS] token and the probabilities for each token on top of them.

Since intent detection and slot filling are classification tasks, we fine-tune the multilingual encoder model XLM-RoBERTa large (Conneau et al., 2019). This allows us to take advantage of cross-lingual transfer by training on different combinations of languages from xSID.

The multitask loss is calculated as the weighted sum of the loss for intent and slot detection

[3]As of version 0.6, the latest version to date, it is available in 17 languages.

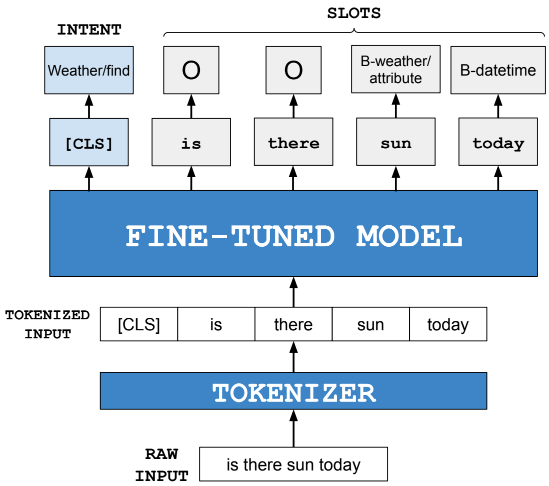

Figure 1: The idea behind the multitask model fine-tuned for both intent detection and slot filling tasks at the same time.

$$\mathcal{L}_{total} = \mathcal{L}_{slot} * \lambda + \mathcal{L}_{intent} * (1 - \lambda) \quad (1)$$

where $\mathcal{L}_{slot}$ is the cross-entropy loss function used in the slot-filling and $\mathcal{L}_{intent}$ is the cross-entropy loss function used in intent-detection. We set λ to 0.7 based on the intuition that slot filling is more difficult. The rest of the fine-tuning hyperparameters can be found in Appendix A. During our experiments, all models have been evaluated using the Norwegian development set, which has been used to select the best combinations of languages and number of epochs.

3.3 Results

We have performed preliminary experiments on the development dataset to select the best combination of languages using three different random seeds. The results of these experiments can be seen in Table 3, where we report the F_1 and accuracy metrics for the slots and intents respectively [4]. We also calculate the Lambda average metric, that is a weighted average, where we use the same λ value as in the multitask loss function.

The results show that training only on the English training data produces the best results, with a Lambda average of 84.96%, probably because machine-translated data can introduce noise to the model.

[4]During the preliminary experiments on the development split, we have used a different scorer than the one provided by the Shared Task. Our scorer uses the output data of the model without post-processing, that allows us to calculate the scores while training.

Language	F_1 Slot	Accuracy Intent	Lambda
EN	**79.09** ±0.77	98.64 ±0.23	**84.96** ±0.48
DA	53.75 ±0.35	98.82 ±0.56	67.04 ±0.05
NB	53.49 ±1.70	98.87 ±0.39	67.10 ±1.16
EN+DA	57.03 ±0.48	98.60 ±0.14	69.50 ±0.30
EN+NB	55.43 ±0.24	**99.17** ±0.13	68.55 ±0.20
DA+NB	54.58 ±0.32	98.94 ±0.21	67.89 ±0.18
EN+DA+NB	58.33 ±1.85	98.73 ±0.25	70.45 ±1.35
ALL	59.83 ±1.88	98.67 ±0.39	71.48 ±1.22
ALL-NB	59.08 ±0.83	98.55 ±0.41	70.92 ±0.66
GER	58.01 ±0.72	98.80 ±0.13	70.25 ±0.47
GER-NB	61.96 ±1.25	98.25 ±0.36	72.85 ±0.98
LAT	58.51 ±0.45	98.80 ±0.37	70.60 ±0.21
LAT-NB	59.62 ±1.34	98.56 ±0.42	71.30 ±1.04

Table 3: F_1 score in the development set for each training language combination, labeling the tokenized sentences. ISO 639-1 Language Codes are used for individual languages, while ALL means the combination of all available training languages, GER means Germanic languages, and LAT means languages written in Latin script. We also sometimes remove Norwegian, e.g., GER-NB would be all Germanic languages except Norwegian Bokmål (full explanation in Appendix B).

	Single-task	Multitask
Slot F_1	78.98 ±0.28	**79.09**±0.77
Intent Accuracy	98.42±0.31	**98.64**±0.23

Table 4: Comparison between the single-task models and the multitask one.

Table 4 compares the multitask and single-task slot-filling and intent classification models, trained in English with the same hyperparameters. We see that not only is multitask training more efficient than single-task training, but it is also able to maintain a similar or slightly better performance (0.11% and 0.22% higher Slot F_1 and Intent accuracy respectively).

For the participation in the shared task, we submit three models: a) the model fine-tuned only on English data b) the model fine-tuned with a combination of English and Norwegian, which obtained the best accuracy for the intent task (99.17%), and c) the model fine-tuned with the combination of all Germanic languages (that have an available training set) minus Norwegian, which obtained the second best results overall (72.85% Lambda average).

The test results are shown in Table 5. Consistent with the evaluation of our models with the development set, the best resulting model is the one trained only using English, with a Lambda average of 88.65%.

Model	Slot F_1	Accuracy Intent	Lambda
EN	**85.37**	96.29	**88.65**
GER-NB	66.64	97.11	75.78
EN+NB	55.66	**97.69**	68.27

Table 5: Results (slot F_1, accuracy intent, and lambda average) of the three submitted runs evaluated on the test set. Best results in bold.

3.3.1 Analysis of the Intent Detection Results

Without any hyperparameter tuning, most models obtain near 100% accuracy in the intent detection task. This is likely because the data is from a reduced domain, where instances contain clear word-level features that let the model infer the label.

To test this idea, we fine-tuned and compared the results of English only models, [BERT[5] (Devlin et al., 2019; Turc et al., 2019) and RoBERTa (Liu et al., 2019)], against multilingual and Norwegian models, [XLM-RoBERTa (Conneau et al., 2019) and NorBERT3 (Samuel et al., 2023)]. Figure 2 shows that no prior knowledge of Norwegian is required to obtain an accuracy of up to 96%, which is aligned with our initial presumption that models are learning to classify the instances relying on specific word patterns rather than semantic understanding. However, prior knowledge of Norwegian greatly reduces the number of parameters required to obtain top performance and allows the model to surpass the performance of English only models.

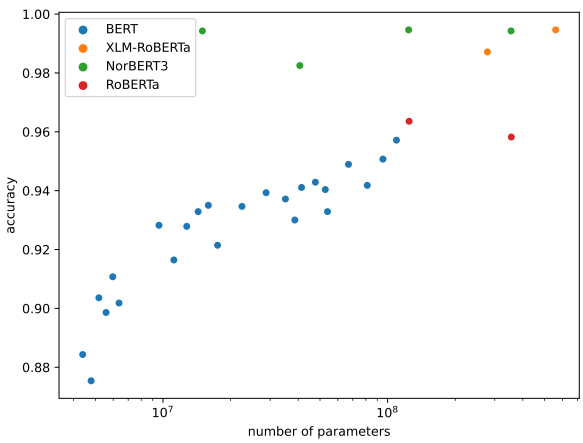

Figure 2: Accuracy of pretrained English models (BERT, RoBERTa), multilingual models (XLM-RoBERTa) and a Norwegian pretrained models (NorBERT3) trained for Intent Detection on the Norwegian train set and evaluated the development set.

[5] Google's 2020 BERT models were fine-tuned.

4 Dialect Identification

In this section, we will present the dialect identification task, starting with the data used in our experiments (Section 4.1), followed by the experimental setting training only on the development set from the shared task (Section 4.2), as well as the experimental settings when training on alternative sources of data (Section 4.3). Finally, we describe the results of using different data and settings (Section 4.4).

4.1 Data

The following section presents all the datasets we have used in our experiments, which consist of the NoMusic data (Table 2) , as well as some further dialectal data. This data comes from two main sources: (i) tweets, which we collected from NorDial and the Nordic Tweet Stream (NTS); and (ii) transcriptions, which come from NB Samtale and the Nordic Dialect Corpus (NDC).

4.1.1 NoMusic

As introduced in Section 2.1, NoMusic is the development data provided by the shared task. However, there is no additional training data that has been labeled for the dialect identification task in the SID tasks.

Consequently, we split the development set into train, development and test sets (from now on, dev-train, dev-dev and dev-test). Each sentence in this dataset is paraphrased 11 times, once for each dialect annotator. Thus, in order to avoid data contamination, we split by the original ID of each instance, as many translated instances are similar or identical (Table 6). The results presented in Section 4.2 correspond to the dev-test results.

Dialect	Dev-Train	Dev-Dev	Dev-Test
West Norwegian (V)	962	220	225
Trøndersk (T)	580	132	135
North Norwegian (N)	386	89	89
Bokmål (B)	188	43	45
Total	2116	484	494

Table 6: Distribution of splits in the development set.

4.1.2 NorDial

NorDial (Barnes et al., 2021) is a corpus of 1,073 Norwegian tweets annotated for four dialects: Bokmål, Nynorsk, Dialect, or Mixed. We merge this data together with the additional annotated data available in the Nordial GitHub,[6] for a total of 6,670 tweets. Table 7 shows the statistics for the merged data.

Split	Train	Dev	Test	Total
Bokmål	2798	115	98	3011
Nynorsk	964	38	43	1045
Dialect	2007	61	70	2138
Mixed	445	12	19	476
Total	6214	226	230	6670

Table 7: Distribution of Norwegian language variants across NorDial splits.

4.1.3 Nordic Tweet Stream (NTS)

NTS[7] (Laitinen et al., 2018) is a corpus of geolocated tweets and their associated metadata from the Nordic region between the years of 2013-2023. We downloaded 4.054.223 Norwegian tweets geolocated in a total of 426 Norwegian cities.

4.1.4 NB Samtale

NB Samtale[8] is a speech corpus collected by the Language Bank at the National Library of Norway. It contains orthographic and verbatim transcriptions from podcasts and recordings of live events at the National Library, a total of 24 hours of transcribed speech from 69 speakers, divided into train, development and test splits. Table 8 shows the distribution of dialects in the data.

Dialect area	Train	Dev	Test	Total
Eastern (E)	4454	557	557	5568
Northern (N)	2072	258	261	2591
Southwest (SW)	1304	164	163	1631
Western (W)	1094	137	136	1367
Central (T)	624	78	78	780
Total	9548	1194	1195	11937

Table 8: Distribution of Norwegian language variants in NB Samtale.

4.1.5 Nordic Dialect Corpus (NDC)

NDC[9] (Johannessen et al., 2009, 2012) includes orthographic and phonetic transcriptions of Nordic speaker recordings, with almost two million words from Norwegian dialects. It contains recordings from 111 different locations in Norway, collected between 2006-2010.

[6]https://github.com/jerbarnes/nordial

[7]https://nordictweetstream.fi/

[8]https://huggingface.co/datasets/Sprakbanken/nb_samtale

[9]https://tekstlab.uio.no/scandiasyn/download.html

4.2 Experiments With Development Data

In this section, we describe baselines using only the dialectal data in the development set, using the splits described in Section 4.1.1 (Table 6). We explore lexical mapping SVM, fine-tuning encoders and decoders, as well as using few-shot decoders.

4.2.1 Lexical Mapping SVM

We first create a simple baseline by mapping common lexical items in Bokmål to their respective dialectal counterparts. The items we map are mainly pronouns and interrogatives, as well as a few common prepositions, verbal forms, and time expressions. For each dialect, there is often a one-to-many mapping from Bokmål, as can be seen in Table 9.

B	V	T	N	EN
jeg	eg, ej	æ, e	æ, å	‘I’
hva	ka	ka	ka	‘what’

Table 9: Example of lexical mappings for B, V, T, N. The English translation is added in the final column.

After compiling the lexical mappings, we create a silver dataset (**Lexmap**) starting from the Bokmål train data provided. Specifically, we create a new instance each for V, T, and N by mapping any lexical item in our mapping dictionary to its dialectal variant, leading to a training dataset four times the size of the original.

We train a linear support-vector machine on unigram features using the silver train set (**Lexmap SVM**). We also train the same model on the silver train plus the dev-train data (**Lexmap + dev-train SVM**).

4.2.2 Encoder Fine-tuning

We fine-tune encoders on the **dev-train** set, as well as on the combination of dev-train with the lexical mapping silver (**Lexmap + dev-train**). We choose the best encoder model specifically trained for Norwegian, NorBERT3-L (Samuel et al., 2023), as well as the multilingual encoder model XLM-Roberta-large (Conneau et al., 2019).[10] As preliminary experiments showed training on the full development set with NorBERT3-L leads to the best performance, we also train the following variants: (i) training on the combined dev-train and

[10]Hyperparameters used are listed in Appendix A.

	NDC-20 (ortho)		NDC-20 (phonetic)		NDC-40 (ortho)		NDC-40 (phonetic)	
Dialect	#	%	#	%	#	%	#	%
West (V)	31017	30.34	35636	30.84	21413	30.74	25803	31.28
North (N)	31387	30.70	34437	29.80	21487	30.85	24459	29.65
Trøndersk (T)	12076	11.81	13502	11.68	7989	11.47	9373	11.36
Bokmål (B)	27763	27.15	31990	27.68	18751	26.93	22868	27.72
Total	102243	100	115565	100	69640	100	82503	100

Table 10: Distribution of dialects in NDC, using a manual geolocation-based mapping of dialect labels, with a minimum token length of 20 and 40 per sentences

	NDC-20 (ortho)		NDC-20 (phonetic)		NDC-40 (ortho)		NDC-40 (phonetic)		NTS	
Dialect	#	%	#	%	#	%	#	%	#	%
West (V)	6891	61.83	36418	37.24	5048	69.62	27845	38.72	49801	46.96
North (N)	52	0.47	218	0.22	33	0.46	45	0.06	16632	15.68
Trøndersk (T)	3877	34.79	60995	62.37	1976	27.25	43909	61.06	30007	28.30
Bokmål (B)	325	2.92	163	0.17	194	2.68	115	0.16	9609	9.06
Total	11145	100	97794	100	7251	100	71914	100	106049	100

Table 11: Distribution of dialects in NDC transcription and NTS tweet datasets, using automatic annotation of dialect labels and dropping instances to match the development distribution.

dev-dev splits (**Dev-train-dev**); and (ii) training on the whole development set (**Dev-train-dev-test**).

4.2.3 Decoder Few-shot

We perform few-shot prompting experiments, providing the model 4 example instances, one for each dialect label. The few-shot examples are sampled from the dev-dev split and we evaluate on the dev-test set. We experiment with a decoder model specifically trained for Norwegian, NorMistral-7b-warm,[11] and a multilingual decoder model, Llama 3.1-8B (Dubey et al., 2024), and use both base and instruct models, evaluating with LM evaluation Harness (Gao et al., 2023). The prompt used in these experiments is shown below:

```
In which dialect is this text written? Choose between North Norwegian, Trøndersk, West Norwegian or Bokmål. Text: {text} Dialect:
```

4.2.4 Decoder Fine-tuning

Next, we fine-tune several decoders on the development set, similar to the experiments with decoders. We only experiment with NorMistral models, as they achieve higher results in few-shot evaluation. We perform finetuning in two ways: by adding a sequence classification (SC) head and training the models applying supervised fine-tuning (SFT) using the same English prompt as in the few-shot evaluation (**dev-train SFT**).

4.3 Experiments With Other Data Sources

As no labeled training dataset is available for dialect classification, we also explore whether it is possible to use other sources of data to learn to classify Norwegian dialects.

First, we apply the semi-automatic and automatic annotation methods (see subsections 4.3.1 & 4.3.2), and get statistics about the resulting dialectal distribution of tweets and transcriptions.

Next, we fine-tune NorBERT3-L on the semi-automatically and automatically labelled transcriptions and tweets to measure the impact of using automatically labeled data sources. During training, we use the dev-dev split as validation to avoid overfitting on these datasets and use the same hyperparameters (see Appendix A).

4.3.1 Semi-automatic Annotation

We perform a semi-automatic dialect label annotation on the NDC dataset, by first eliminating special transcription characters, e.g., pause markers (#) or (mm), as well as short sentences, which we assume have fewer dialectal traits.[12]

Finally, we semi-automatically map cities in NDC to their corresponding dialect label, according to their geographical location.[13] Table 10 re-

[11] https://huggingface.co/norallm/normistral-7b-warm

[12] We experiment with two different minimum sentence lengths: 20 and 40 tokens.

[13] Eastern cities are mapped to Bokmål.

ports the number of instances and the dialect label distribution.

4.3.2 Automatic Annotation

We automatically annotate silver training data using two classifiers: the best model trained on development data (see Section 4.4.1) and a model trained on NorDial data. Experiments on NorDial suggest NB-BERT-base is the strongest classifier, achieving 90% weighted F_1 score, thus being chosen as our NorDial classifier. The objective of using two classifiers is to minimize model bias.

Therefore, having the results of our two classifiers, we discard examples classified as Nynorsk and Mixed by the NorDial classifier. For Bokmål, we select examples where the two classifiers match. For the dialectal tweets, we assign the class of the NorBERT3-L classifier if it is one of N, V or T.

NB Samtale We train a classifier on NB Samtale data with the available splits to measure to what extent there are dialectal features in the orthographic and verbatim transcriptions. We get a weighted F_1 of 76.76% with the verbatim transcriptions, so we can conclude that the models are able to learn the different features of the dataset. However, as training on this data leads to poor results on the dev set, we decide to explore other annotation methods. The poor results suggest that the dialectal features present in both datasets are different. Additionally, we trained a model using both NB Samtale train set and dev-train, but the results obtained (F_1 81.59%) are few points worse than the model trained only in dev-train (F_1 82.44%).

NTS The predicted distribution of dialects in NTS tweets does not match with the Norsid classifier distribution. Nordial classifier classifies 96.70% of instances as Bokmål and Norsid classifier 66.93% as V. This makes sense because the distributions of their training data are different. After performing the automatic labeling, in order to obtain a distribution similar to the one we have in development, we have downsampled the automatically-labelled NTS instances until the distribution matches that of development (see Table 11).

NDC We have additionally automatically annotated the NDC instances (see Table 11). In most cases, there is a large difference between semi-automatic and automatic labeling. This could be due to the training data for our classifier differing from the instances in the NDC dataset, but we decided to follow the same annotation approach in order for the results to be comparable. Moreover, it is important to note that the automatic labeling distribution does not match the development set distribution; thus, our procedure has a bias toward annotating instances as V or T. The dialect identification results when using data annotated with this approach obtains better results than semi-automatic annotation and NB Samtale (see Table 12), so we apply this classification method to the following dataset annotations.

Dataset	Model	Dev F_1	Test F_1
-	Majority Random	28.10 30.38	27.67 32.40
Lexmap Lexmap + dev-train	SVM	53.91 66.98	56.11 70.02
Dev-train	XLM-R-L NorBERT3-L	61.85 **82.44**	63.76 82.71
Lexmap + dev-train	NorBERT3-L	75.85	75.32
Dev-train-dev Dev-train-dev-test	NorBERT3-L	- -	**84.17** 83.34
Dev few-shot	NorMistral-7b NorMistral-7b-it Llama3.1-8B Llama3.1-8B-it	29.69 38.24 28.65 28.64	29.55 30.83 30.12 28.88
Dev-train	NorMistral-7b (SC) NorMistral-7b (SFT) NorMistral-7b-it (SFT)	78.69 76.79 76.43	74.91 76.88 74.16
NTS*	NorBERT3-L	64.60	64.22
NDC-20-orth* NDC-40-orth* NDC-20-phon* NDC-40-phon*	NorBERT3-L	33.65 34.31 51.23 48.26	34.10 33.82 52.09 48.50
NDC-20-orth† NDC-40-orth† NDC-20-phon† NDC-40-phon†	NorBERT3-L	36.02 32.08 44.40 44.97	36.05 35.39 44.15 43.78
NB Samt NB Samt + Dev-train	NorBERT3-L	32.45 81.59	30.48 81.76

Table 12: Weighted F_1 results of Dialect Identification subtask. * refers to the dataset annotated automatically and † to semi-automatically. *it* refers to the instruct version of the models and *L* the large version of the models.

4.4 Results

The results were calculated using the official evaluation script of the shared task and the official metric, Weighted F_1 Score. All dev results in this section correspond to dev-test.

4.4.1 Training Only on Development Data

The lexical mapping baseline performs better than majority or random, achieving 53.91 and 56.11 weighted F_1 on the dev-test and test sets, respectively. Further training on the dev-train set improves this to 66.98 and 70.02.

There is a large difference between the two encoder models (see Table 12). Whereas XLM-Roberta does not reach the best lexical mapping baseline, NorBERT3-L surpasses the Lexmap + dev-train baseline by 15.46 points on the development set. Additionally training with the Lexmap data, however, harms performance by 7 points. NorBERT3-L models trained in Dev-train-dev and Dev-train-dev-test obtain the highest results the test set.

In the few-shot scenario, the four models barely beat the majority class baseline (27.67) and perform worse than a random classifier (32.82). NorMistral Instruct (30.83) is slightly better than its base counterpart (29.55), but they are still far from the lexical mapping baseline, which obtains around 30 points more. Regarding Llama3.1 base and instruct models, their performance is almost identical to NorMistral models, but none of them surpass the performance of NorMistral Instruct in this few-shot evaluation. Fine-tuning NorMistral gives better results than the few-shot approach (76.88).

4.4.2 Training on Other Sources of Data

The results in Table 12 suggest that using tweets is better than transcriptions, in both semi-automatically and automatically labeled experiments: we obtain a weighted F_1 of 64.22 in our tweets model, while the transcription models perform between 30-52 points. However, the performance of the tweets model is still far from models trained on the development set (84.17).

When using transcriptions, the phonetic ones are preferable to orthographic ones, as more dialectal features are retained. Using longer sentences (>40 tokens) generally has little impact on performance, except for automatically labeled phonetic transcriptions.

The model trained on NB Samtale dataset achieves lower scores than models trained on NDC and NTS. This seems to be due to a low overlap in dialectal features between the NB Samtale and the shared task data.

4.4.3 Dialect Analysis

We have selected the best performing models from each strategy to analyze the performance in each dialect. The models we have chosen are, Dev-train-dev NorBERT3-L, Few-shot NorMistral-7b-warm-it, NTS NorBERT3-L, Semi-automatic labeled NDC-20-phon NorBERT3-L and Automatic labeled NDC-20-phon NorBERT3-L (see Table 13).

For the best models trained on dev (NorBERT3-L and NorMistral-7b-warm (SFT)) the label imbalance affects performance, with models performing better on labels with more examples. We see this same pattern in the tweets dataset, as the dialect label distribution in the NTS dataset is similar to the one in the development set. For semi-automatic transcriptions, a higher performance is also observed on the majority classes, with the exception of Bokmål, probably due to annotation errors. In the automatic transcription datasets, the class imbalance is even larger, and this is reflected in even worse results for the minority classes. Finally, we see that the few-shot decoder model has a bias for T, as it assigns the other labels less often.

Dataset	Model	B	N	T	V
Dev-train-dev	NorBERT3-L	74.10	75.72	83.97	86.61
Few-shot	NorMistral-7b-it	06.56	00.88	42.12	12.87
Dev-train	NorMistral-7b (SFT)	71.48	71.65	83.07	76.13
NTS	NorBERT3-L	55.83	50.29	60.17	71.39
NDC-20-phon†	NorBERT3-L	14.17	39.73	19.95	58.75
NDC-20-phon*	NorBERT3-L	31.09	06.62	52.91	69.24

Table 13: Test F_1 per dialect with the best performing models in each category. *it* refers to the instruct version of the models and *L* the large version of the models.

5 Conclusion and Future Work

We have presented our submission for the NorSID Shared Task in the 2025 VarDial Workshop (Scherrer et al., 2025). We have participated in the three proposed tasks – Intent Detection, Slot Filling and Dialect Identification – with 3 submissions for each of them.

For the Intent Detection & Slot Filling tasks we designed a multitask model, improving efficiency with respect to having a model for each task. Additionally, as both tasks are highly related, this combination improves the performance of the model in both tasks to 97.69% accuracy and 85.37% F_1, respectively, in the test set.

In Dialect Identification, we tested many different approaches by using the development data as training, as well as additional data from tweets and transcriptions. However, none of the settings we tried were able to surpass the performance of NorBERT3-L fine-tuned only on the development set, which achieved 84.17 F_1 on the test set.

The research presented in this paper has opened the way to many questions that need further investigation. We believe that the results could be improved using better encoder, e.g., DeBERTa (He et al., 2021), and decoder, e.g., Llama 3.1 70B) models. The additional data we collected for dialect identification has not been successful due to the narrow domain of the tasks, but it is likely that for other tasks with a stronger domain shift this data could provide for more robust training.

Acknowledgements

This work has been partially funded by:

- Disargue (TED2021-130810B-C21) MCIN/AEI/10.13039/501100011033 and European Union NextGenerationEU/PRTR.
- DeepKnowledge (PID2021-127777OB-C21) MCIN/AEI/10.13039/501100011033 and by FEDER, EU.
- Ixa group A type research group (IT1570-22) Basque Government
- IKER-GAITU project 11:4711:23:410:23/0808 by Basque Government
- Julen Etxaniz holds a PhD grant from the Basque Government (PRE_2024_2_0028).
- Maite Heredia is supported by the UPV/EHU PIF23/218 predoctoral grant.
- The EU Horizon Europe Framework under the grant 101135724 (LUMINOUS).

References

Noëmi Aepli, Çağrı Çöltekin, Rob van der Goot, Tommi Jauhiainen, Mourhaf Kazzaz, Nikola Ljubešić, Kai North, Barbara Plank, Yves Scherrer, and Marcos Zampieri. 2023. Findings of the VarDial evaluation campaign 2023. In *Proceedings of the Tenth Workshop on NLP for Similar Languages, Varieties and Dialects*, Dubrovnik, Croatia. Association for Computational Linguistics.

Jeremy Barnes, Petter Mæhlum, and Samia Touileb. 2021. Nordial: A preliminary corpus of written norwegian dialect use. *NoDaLiDa 2021*, page 445.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *Preprint*, arXiv:1805.10190.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Drew Harwell. 2018. The accent gap. *The Washington Post*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. *Preprint*, arXiv:2006.03654.

Janne Bondi Johannessen, Joel Priestley, Kristin Hagen, Tor Anders Åfarli, and Øystein Alexander Vangsnes. 2009. The nordic dialect corpus–an advanced research tool. In *Proceedings of the 17th nordic conference of computational linguistics (nodalida 2009)*, pages 73–80.

Janne Bondi Johannessen, Joel Priestley, Kristin Hagen, Anders Nøklestad, and André Lynum. 2012. The nordic dialect corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3387–3391.

Mikko Laitinen, Jonas Lundberg, Magnus Levin, and Rafael Messias Martins. 2018. The nordic tweet stream: A dynamic real-time monitor corpus of big and rich language data. In *Digital Humanities in the Nordic Countries 3rd Conference, Helsinki, Finland, March 7-9, 2018*, pages 349–362. CEUR-WS. org.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019.

Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Petter Mæhlum and Yves Scherrer. 2024. NoMusic - the Norwegian multi-dialectal slot and intent detection corpus. In *Proceedings of the Eleventh Workshop on NLP for Similar Languages, Varieties, and Dialects (VarDial 2024)*, pages 107–116, Mexico City, Mexico. Association for Computational Linguistics.

David Samuel, Andrey Kutuzov, Samia Touileb, Erik Velldal, Lilja Øvrelid, Egil Rønningstad, Elina Sigdel, and Anna Palatkina. 2023. NorBench – a benchmark for Norwegian language models. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 618–633, Tórshavn, Faroe Islands. University of Tartu Library.

Yves Scherrer, Rob van der Goot, and Petter Mæhlum. 2025. VarDial evaluation campaign 2025: Norwegian slot and intent detection and dialect identification. In *Proceedings of the Twelfth Workshop on NLP for Similar Languages, Varieties, and Dialects (VarDial 2025)*, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3795–3805, Minneapolis, Minnesota. Association for Computational Linguistics.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*.

Rob van der Goot, Ibrahim Sharaf, Aizhan Imankulova, Ahmet Üstün, Marija Stepanovic, Alan Ramponi, Siti Oryza Khairunnisa, Mamoru Komachi, and Barbara Plank. 2021. From masked-language modeling to translation: Non-English auxiliary tasks improve zero-shot spoken language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Mexico City, Mexico. Association for Computational Linguistics.

Miriam Winkler, Virginija Juozapaityte, Rob van der Goot, and Barbara Plank. 2024. Slot and intent detection resources for Bavarian and Lithuanian: Assessing translations vs natural queries to digital assistants. In *Proceedings of The 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*. Association for Computational Linguistics.

A Hyperparameters

A.1 Slot-intent Multitask Model

The hyperparameters used in the slot-intent multitask model are the following:

- *Learning rate:* $2e^{-5}$
- *Batch size:* 64
- *Number of epochs:* 10
- *Weight decay:* 0.01

A.2 Dialect Detection Model

The hyperparameters used in dialect classification task are the following:

NorBERT3-L:

- *Learning rate:* $5e^{-5}$
- *Batch size:* 16
- *Number of epochs:* 15
- *Weight decay:* $1e^{-4}$

XLM-RoBERTa-L:

- *Learning rate:* $1e^{-5}$
- *Batch size:* 16
- *Number of epochs:* 15
- *Weight decay:* $1e^{-4}$

NorMistral:

- *Learning rate:* $5e^{-5}$
- *Batch size:* 16
- *Number of epochs:* 5
- *Weight decay:* $1e^{-4}$

B Languages Combination

The language combinations used in the slot-intent multitask model are the next ones:

- English (EN): Only the English language. This language is the only one that is not machine-translated in the xSID dataset
- Danish (DA): Only the Danish language. This language is the closest language to Norwegian in the xSID dataset.
- Norwegian (NB): Only the Norwegian training data provided. This data is poorly machine-translated, because of this, it was excluded from some combination of languages.
- English and Danish (EN+DA): The combination of English and Danish languages.
- English and Norwegian (EN+NB): The combination of English and Norwegian languages.
- Danish and Norwegian (DA+NB): The combination of Danish and Norwegian languages.
- English, Danish, and Norwegian (EN+DA+NB): The combination of English, Danish, and Norwegian.
- All languages (ALL): All languages on the xSID dataset (Arabic Danish German English Indonesian Italian Japanese Kazakh Dutch Serbian Turkish Chinese) and the Norwegian data provided.
- All languages without Norwegian (ALL-NB): All languages on the xSID dataset (Arabic Danish German English Indonesian Italian Japanese Kazakh Dutch Serbian Turkish Chinese).
- Germanic languages (GER): Germanic languages on the xSID dataset (Danish German English Dutch) and the Norwegian data provided.
- Germanic languages without Norwegian (GER-NB): Germanic languages on the xSID dataset (Danish German English Dutch)
- Latin script languages (LAT): Languages that have latin script in the xSID dataset (Danish German English Indonesian Italian Dutch Serbian Turkish) and Norwegian.
- Latin script languages without Norwegian (LAT-NB): Languages that have latin script in the xSID dataset (Danish German English Indonesian Italian Dutch Serbian Turkish).