Improving Dialectal Slot and Intent Detection with Auxiliary Tasks: A Multi-Dialectal Bavarian Case Study

Xaver Maria Krückl* **Verena Blaschke*** **Barbara Plank**
MaiNLP, Center for Information and Language Processing, LMU Munich, Germany
Munich Center for Machine Learning (MCML), Munich, Germany
xaver.krueckl@gmail.com, {verena.blaschke, b.plank}@lmu.de

Abstract

Reliable slot and intent detection (SID) is crucial in natural language understanding for applications like digital assistants. Encoder-only transformer models fine-tuned on high-resource languages generally perform well on SID. However, they struggle with dialectal data, where no standardized form exists and training data is scarce and costly to produce. We explore zero-shot transfer learning for SID, focusing on multiple Bavarian dialects, for which we release a new dataset for the Munich dialect. We evaluate models trained on auxiliary tasks in Bavarian, and compare joint multi-task learning with intermediate-task training. We also compare three types of auxiliary tasks: token-level syntactic tasks, named entity recognition (NER), and language modelling. We find that the included auxiliary tasks have a more positive effect on slot filling than intent classification (with NER having the most positive effect), and that intermediate-task training yields more consistent performance gains. Our best-performing approach improves intent classification performance on Bavarian dialects by 5.1 and slot filling F1 by 8.4 percentage points.

1 Introduction

Most research on natural language processing (NLP) for digital assistants has focused on standardized languages, despite the large degree of dialectal variation exhibited by many languages and the positive attitude towards dialectal versions of such technologies expressed by some speaker communities (Blaschke et al., 2024b).

A core task of natural language understanding (NLU) is to detect the intent of an input to a digital assistant (e.g., the instruction "delete all alarms" belongs to the *cancel alarm* class) and to tag it for specific slots (e.g., "all" should be tagged as the *reference* associated with the intent). However, classifying dialectal inputs is still challenging as contemporary models are less proficient due to the scarcity of low-resource and especially dialectal training data (Zampieri et al., 2020). To overcome this issue, transferring task knowledge cross-lingually from high-resource language data to low-resource varieties is a strategy widely used in NLU (Upadhyay et al., 2018; Schuster et al., 2019a; Xu et al., 2020, inter alia). While many approaches have focused on cross-lingual transfer via embedding transmission and machine translation, van der Goot et al. (2021a) use non-English auxiliary task data for zero-shot transfer to other languages.

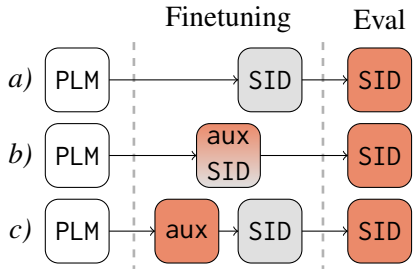

Figure 1: **Overview of evaluated setups.** We fine-tune pre-trained language models (PLMs) on English SID data (grey) and evaluate them on Bavarian (red). We compare multiple setups: *a)* no auxiliary tasks, *b)* multi-task learning by jointly training on English SID data and Bavarian auxiliary tasks ("aux"), *c)* intermediate-task training on Bavarian, then fine-tuning on English SID data.

Inspired by this setup and by intermediate-task training procedures (Pruksachatkun et al., 2020), we use auxiliary tasks to analyze and improve zero-shot transfer learning for slot and intent detection (SID) for Bavarian dialects (Figure 1). To account for intra-dialectal variation, we evaluate on two previously released Bavarian datasets and introduce a third test set. For the auxiliary tasks, we use three recent Bavarian datasets for syntactic annotations, named entity recognition (NER), and masked language modelling (MLM).

*Equal contribution.

Proceedings of the Twelfth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 128–146
January 19, 2025.

We make the following contributions:

- We release a new Bavarian slot and intent detection evaluation dataset (§4.1).[1]
- We examine how training on auxiliary NLP tasks in Bavarian affects SID performance (§6.2). We compare both the integration of the auxiliary tasks into the training setup (joint multi-task learning vs. intermediate-task training) and the tasks themselves.
- To analyze the robustness of the results, we examine performance and data differences between the dialectal test sets (§6.3, 6.4) and include additional datasets (§6.5).

We share our code publicly.[2]

2 Related Work

Slot and intent detection for dialects and non-standard varieties Research on SID for low-resource languages, including non-standard and dialectal varieties, has started receiving more attention. This trend starts with van der Goot et al. (2021a), who introduce a multilingual SID dataset, xSID, containing South Tyrolean, a Bavarian dialect (more details in §4.1). xSID has since been extended with dialectal data from Upper Bavaria (Winkler et al., 2024), data in Bernese Swiss German and Neapolitan (Aepli et al., 2023), and eight Norwegian dialects (Mæhlum and Scherrer, 2024).

Similarly to our study, van der Goot et al. (2021a) experiment with multi-task learning, although they only have Standard German auxiliary data at their disposal for the South Tyrolean test data. Other approaches focus on tokenization issues or data augmentation. Srivastava and Chiang (2023) tackle tokenization issues caused by spelling differences by injecting character-level noise into standard-language training data, which improves the performance on the dialectal test sets. Muñoz-Ortiz et al. (2025) find that encoding text with visual representations (rather than ones based on subword tokens) improves transfer from Standard German to German dialects for intent classification. Abboud and Oz (2024) fine-tune a masked language model on dialectal data to generate synthetic training data for German and Arabic dialects. Malaysha et al. (2024) organized a shared task on intent detection in four Arabic dialects, where the top systems all involve model ensembling and translating the training data into the test dialects (Ramadan et al., 2024; Elkordi et al., 2024; Fares and Touileb, 2024).

In the context of spoken intent classification, other work focuses on variation in spoken Italian (Koudounas et al., 2023) and English (Gerz et al., 2021; Rajaa et al., 2022; He and Garner, 2023).

Multi-task learning (MTL) Joint MTL learning involves jointly training a model on several tasks. Ruder (2017) provides a general overview. Martínez Alonso and Plank (2017) find that tasks with non-skewed label distributions lend themselves best as auxiliary tasks for sequence tagging. Schröder and Biemann (2020) show that auxiliary tasks which are more similar to the target tasks result in better target performance.

Regarding MTL for SID, Wang et al. (2021) train a transformer model on dependency parsing, POS tagging, and SID, with different layers attending to the different tasks. They find that the syntactic tasks improve SID performance (especially when both are included), and that jointly producing slot and intent labels is also beneficial.

Van der Goot et al. (2021a) use English training data for SID but additionally exploit non-English auxiliary task data, hypothesizing that this helps their models to learn additional linguistic properties of the target language. They find syntactic tasks to be useful for slot filling for one pre-trained language model but not another, and harmful for intent detection. Similarly, they find masked language modelling (MLM) to be of use for slot filling but not intent classification. Machine translation as auxiliary task yielded worse performance.

Intermediate-task training While MTL is about fine-tuning a model *simultaneously* on multiple tasks, intermediate-task training concerns first fine-tuning a model on one or more auxiliary tasks and *subsequently* fine-tuning it on the target task. In a similar vein to some MTL results, Poth et al. (2021) and Padmakumar et al. (2022) find the similarity between the intermediate and target task to be important. Similarly, Pruksachatkun et al. (2020) evaluate models on inference and reading understanding tasks and find including intermediate tasks also related to reasoning to be useful. Padmakumar et al. (2022) further find that including multiple intermediate tasks at once often yields better results than only including one, although the interactions

[1] To be included in https://github.com/mainlp/xsid.

[2] https://github.com/mainlp/auxtasks-bavarian-sid

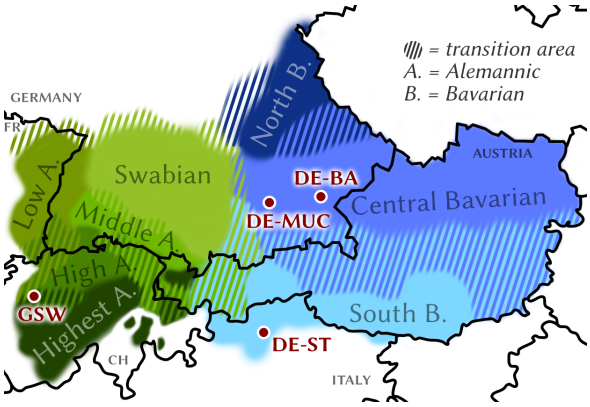

Figure 2: **The Upper German dialect groups Bavarian** (blue, right) **and Alemannic** (green, left), based on Wiesinger (1983). The red dots show the xSID datasets included in this study and our new dataset, de-muc.

of tasks are difficult to predict.

In the context of cross-lingual evaluation, Samuel et al. (2022) find that continued pretraining via target-language MLM has mixed results. Phang et al. (2020) show that even in cross-lingual scenarios, intermediate-task learning on the source language can be beneficial.

Some recent studies include both MTL and intermediate-task training. Weller et al. (2022) find that MTL with several auxiliary tasks tends to perform worse than with just one additional task, and that MTL beats intermediate-task training when the target task has less data than the auxiliary task. Montariol et al. (2022) focus on cross-lingual hate speech detection and add auxiliary tasks in multiple languages (including the target language). They find joint MTL setups to outperform intermediary task training, and semantic auxiliary tasks to be more beneficial than syntactic ones.

3 Background: Bavarian Dialects

Bavarian dialects differ from Standard German in phonetics, phonology, word choice, and morphosyntax (Merkle, 1993). There is no established orthography or standard variety of Bavarian. The Bavarian dialects belong to the Upper German dialect group and are split into three major subgroups (Northern, Central, and Southern Bavarian; Figure 2), mostly based on sound differences (Wiesinger, 1983). There is also phonetic/phonological and lexical variation *within* these groups (Rowley, 2023, passim). The pronunciation differences are also reflected in the spelling choices made in the different training and evaluation datasets in our study, although the spellings also reflect idiosyncratic preferences. We compare the Bavarian SID test sets in §6.4.

Some of the morphosyntactic differences between Bavarian and Standard German (cf. Blaschke et al., 2024a) are relevant for SID, and recent work (Artemova et al., 2024) has shown that slot filling performance in German is negatively affected by dialectal syntactic structures. Person names are typically preceded by definite articles, and the given name generally follows the family name (Weiß, 1998, pp. 69–71) – this has been analyzed in the context of NER (Peng et al., 2024) and might also be relevant for slot filling. Furthermore, many NLU queries contain infinitive constructions of the form "remind me to [do X]". Such cases are often expressed with a nominalized infinitive construction (Bayer, 1993; Bayer and Brandner, 2004; see, e.g., Table 10) that does not exist in Standard German.

Additionally, as in many other German dialects (Weise, 1910), temporal expressions (relevant for `datetime` slots) can be expressed in ways that are not grammatical in Standard German, e.g., *fia fünfe heid auf Nacht* "for 5PM tonight" (lit. "for five today at night") or *um 3 nammiddog* "at 3PM" (lit. "at 3 afternoon").

4 Data

4.1 Slot and Intent Detection Data

xSID We use xSID 0.5 (van der Goot et al., 2021a; CC BY-SA 4.0), which provides development and test sets (300 and 500 sentences, respectively) for slot and intent detection in a range of languages, as well as a large English training set (44k sentences). It covers 16 intents and 33 different slot types. The data consist of re-annotated English sentences from SNIPS (Coucke et al., 2018) and a Facebook dataset (Schuster et al., 2019b). The non-English development and test splits are translations.

xSID 0.5 contains multiple Upper German dialects (Figure 2), none of which are standardized: South Bavarian as spoken in South Tyrol (**de-st**; included in the first xSID release), Central Bavarian as spoken in Upper Bavaria (**de-ba**; Winkler et al., 2024), and Swiss German as spoken in Bern (**gsw**; Aepli et al., 2023). We focus on the Bavarian test sets, but include the Swiss German data as well as the Standard German (**de**) and English (**en**) test sets in an additional evaluation (§6.3).

Munich Bavarian evaluation data To investigate the effect of intra-dialectal variation and differ-

ent translation choices, we create a second Central Bavarian translation. The new test and development set is in the dialect spoken in Munich (**de-muc**), translated by a native speaker (one of the authors). The translation is directly from English, without referencing either the Standard German or dialectal versions, as was also done for the other dialect translations. The (sentence-level) intent labels are the same as in English and the other languages; the (token-level) slot spans were annotated by the translator. As there is no Bavarian orthography, de-muc represents the spelling preferences of the translator. The grapheme–phoneme mapping is similar to that of Standard German and reflects the translator's pronunciation. Most words are lowercased, also nouns that would be capitalized in German. Named entities are left untranslated and, per the xSID guidelines, grammatical mistakes in the original sentences are also adopted in the translations.

Our Munich Bavarian translations are the most similar to the other Central Bavarian ones (de-ba) on a word and character level (see Appendix A).

We share a data statement (Bender and Friedman, 2018) in Appendix B.

Additional evaluation data To evaluate whether some of our findings generalize to other Bavarian datasets, we use test sets provided by Winkler et al. (2024). They collected naturalistic data by asking Bavarian speakers to come up with queries for a digital assistant that match xSID's intents, and translated a subset of MASSIVE (FitzGerald et al., 2023) with the labels mapped to match xSID's. The translator for MASSIVE is the same as for xSID's de-ba set, and the contributors to the naturalistic data also come from the same region.

4.2 Auxiliary Task Data Sets

We use three Bavarian datasets for auxiliary NLP tasks. These tasks are similar to ones explored in related work on MTL for SID (§2) and are additionally motivated by data availability.

Syntactic dependencies and POS tags (UD) As token-level information and linguistic structure might be useful for slot annotations, we include two syntactic tasks: dependency parsing and part-of-speech (POS) tagging. The Universal Dependencies v2.14 (UD; de Marneffe et al., 2021) treebank MaiBaam (Blaschke et al., 2024a; CC BY-SA 4.0) provides such dependency annotations and POS tags for Bavarian dialects from all three Bavarian dialect groups, including varieties spoken in South Tyrol, Upper Bavaria, and Munich. MaiBaam contains some sentences from xSID, which we exclude from our experiments, leaving 975 sentences that we randomly divide into training and development data using a 90:10 split.

Named entity recognition (NER) Similarly to slot filling, NER concerns identifying and labelling spans of tokens as a sequence tagging task. BarNER 1.0 (Peng et al., 2024) provides such annotations for named entities in Wikipedia articles (CC BY-SA 4.0) and tweets. Based on the inspection of a small data sample, Peng et al. state that the most represented Bavarian dialect group is Central Bavarian (to which both de-ba and de-muc belong). We use the predefined training and development splits (9k and 918 sentences, respectively), and use the fine-grained label set.

Masked language modelling (MLM) We also include MLM, as it is a common pre-training objective.[3] We use a subset of the Bavarian Wikipedia (1.5k sentences, divided into 90% training and 10% development data), as pre-processed by Artemova and Plank (2023).

5 Methodology

We fine-tune pre-trained language models (PLMs) on xSID's English training data using MaChAmp 0.4.2 (commit `9f5a6ce`; van der Goot et al., 2021b) with the same hyperparameters as van der Goot et al. (2021a) did for their SID experiments.

We evaluate slot predictions with strict slot F1, intent predictions with accuracy, and also calculate the proportion of sentences with fully correct predictions. We treat SID itself as a multi-task setup as we jointly predict the slots and intent labels, and treat slot detection as a basic sequence labelling task with a final softmax layer. We use the following task types for MaChAmp (van der Goot et al., 2021b): `seq` (slot filling, NER, POS tagging), `classification` (intent classification), `mlm` (MLM), and `dependency` (dependency parsing). The loss for each task is weighted equally. We use MaChAmp's default loss functions (cross-entropy loss for all tasks except dependency parsing, which uses negative log likelihood). We provide mean scores across three runs for each experiment.

[3] We note however that mDeBERTa v.3 is pre-trained on replaced token detection rather than MLM (He et al., 2021a).

We compare three types of experimental setups (Figure 1):

Baseline We compare four commonly used PLMs, which we finetune on SID data without auxiliary tasks: the monolingual German GBERT (Chan et al., 2020), and the multilingual models mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020), and mDeBERTa v.3 (He et al., 2021a,b).

Notably, mBERT's pretraining data also includes the Bavarian Wikipedia, which contains articles in all three of our test dialects. XLM-R and mDeBERTa were pre-trained on the CC-100 dataset (Conneau et al., 2020), which does not contain Bavarian data. GBERT's pretraining data is in Standard German. To limit computation costs, we use the base-sized versions.[4] In the remaining setups, we only use mDeBERTa because of its strong performance as a baseline PLM (§6.1).

Multi-Task Learning We train the model to jointly predict labels for SID and at least one auxiliary task. We use × to denote these setups, e.g., NER×SID refers to training a model to simultaneously predict named entity labels, slots and intents.

Intermediate-Task Training We first train the model to predict labels for an auxiliary task, remove the task-specific head, optionally repeat this for a second auxiliary task, and then finally train the model to predict SID labels. We use → to denote these setups, e.g. NER→SID refers to first training a model on NER data, then on SID data. As a special case, we train some models first jointly on auxiliary tasks and then afterwards on SID (e.g., MLM×NER→SID).

We apply each auxiliary task dataset to both finetuning setups. For the settings with multiple auxiliary tasks, we select combinations that appear promising based on the results already obtained. We were not able to examine all possible combinations due to computational restraints.

6 Results and Analysis

We first present the results of the baseline models (§6.1), and then discuss the impact of finetuning the model on auxiliary Bavarian NLP tasks (§6.2). We next compare performances across

[4]We use deepset/gbert-base (license: MIT), google-bert/bert-base-multilingual-cased (Apache 2.0), FacebookAI/xlm-roberta-base (MIT), and microsoft/mdeberta-v3-base (MIT).

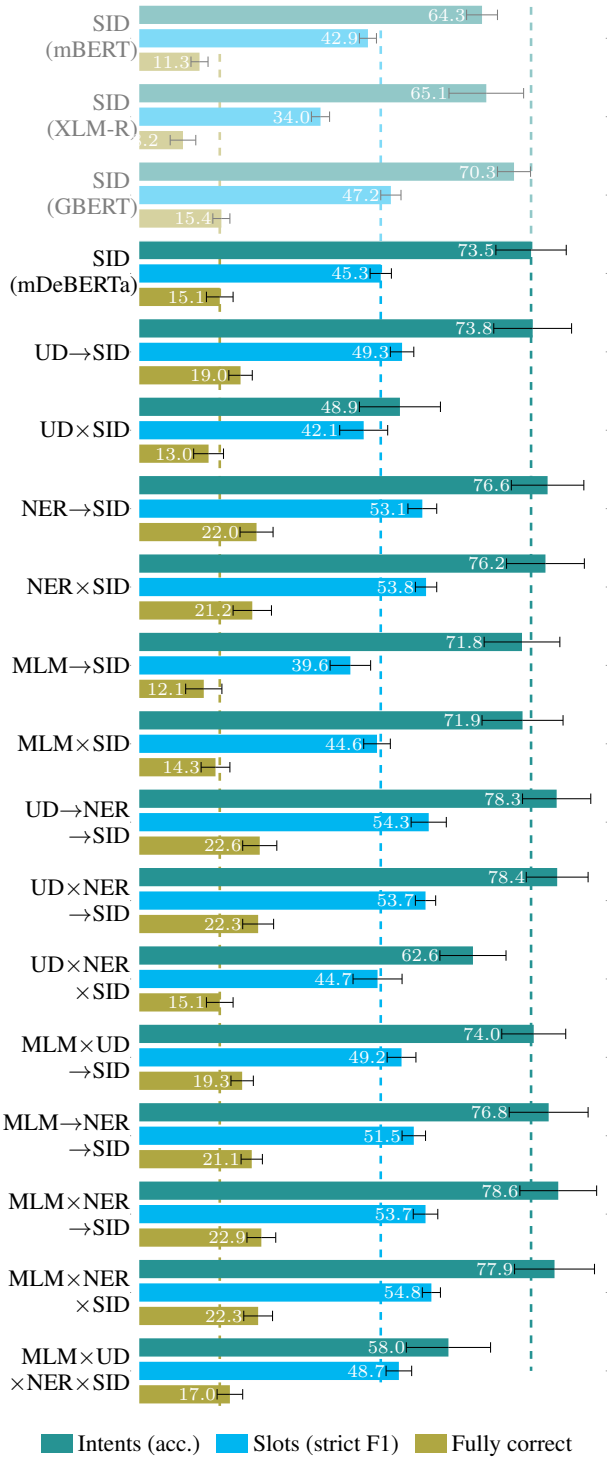

Figure 3: **Slot and intent detection results for the different models, in %.** The results are averaged over the three Bavarian dialect test sets and three random seeds (standard deviations shown as error bars). Mean scores and standard deviations per individual dialect are in Appendix D. The dashed lines denote the scores of the baseline model (no auxiliary tasks). The setups with auxiliary tasks also use mDeBERTa. The three pale entries at the top are worse-performing baseline models with alternative PLMs.

	Intents						Slots					
	Avg	Δ to baseline					Avg	Δ to baseline				
		ITT (→SID)	MTL (×SID)	UD	NER	MLM		ITT (→SID)	MTL (×SID)	UD	NER	MLM
SID (mDeBERTa)	73.5						45.3					
UD→SID	73.8	+0.3		+0.3			49.3	+3.9		+3.9		
UD×SID	48.9		–24.6	–24.6			42.1		–3.2	–3.2		
NER→SID	76.5	+3.0			+3.0		53.1	+7.8			+7.8	
NER×SID	76.2		+2.7		+2.7		53.8		+8.4		+8.4	
MLM→SID	71.8	–1.8				–1.8	39.6	–5.8				–5.8
MLM×SID	71.9		–1.6			–1.6	44.6		–0.7			–0.7
UD→NER→SID	78.3	+4.8		+4.8	+4.8		54.3	+9.0		+9.0	+9.0	
UD×NER→SID	78.4	+4.8		+4.8	+4.8		53.7	+8.4		+8.4	+8.4	
UD×NER×SID	62.6		–10.9	–10.9	–10.9		44.7		–0.6	–0.6	–0.6	
MLM×UD→SID	73.9	+0.4		+0.4		+0.4	49.2	+3.8		+3.8		+3.8
MLM→NER→SID	76.8	+3.3			+3.3	+3.3	51.5	+6.2			+6.2	+6.2
MLM×NER→SID	78.6	+5.1			+5.1	+5.1	53.7	+8.4			+8.4	+8.4
MLM×NER×SID	77.9		+4.3		+4.3	+4.3	54.8		+9.5		+9.5	+9.5
MLM×UD×NER×SID	58.0		–15.6	–15.6	–15.6	–15.6	48.7		+3.3	+3.3	+3.3	+3.3
Mean		+2.5	–7.6	–5.8	+0.2	–0.8		+5.2	+2.8	+3.5	+6.7	+3.5
Std. deviation		2.6	11.4	11.4	7.7	7.1		4.9	5.2	4.4	3.3	5.3

Table 1: **Differences to the baseline performance per set-up type** (intermediate-task training (ITT) vs. MTL) **and auxiliary task** (UD, NER, MLM), in percentage points (pp.). E.g., the results of the intermediate task-training set-up with the UD tasks (UD→SID) beat the baseline by 0.3 pp. for intent detection and 3.9 pp. for slot-filling. Scores are averaged across the Bavarian test sets and three random seeds.

the Bavarian dialects as well as an additional Upper German dialect (Bernese Swiss German) and the standard languages German and English (§6.3). We additionally discuss differences between the Bavarian translations (§6.4), and lastly analyze the results on other Bavarian SID datasets (§6.5).

6.1 Baselines: No Auxiliary Tasks

Our baseline experiments with mBERT and XLM-R achieve similar scores to the results reported by van der Goot et al. (2021a) for the overall cross-lingual xSID test sets (Appendix C). However, these two models perform worse than GBERT and mDeBERTa on the Bavarian test sets (see top of Figure 3). GBERT provides the best slot filling scores (F1: 47.2%) and a slightly higher proportion of fully correctly annotated sentences (15.4%, mDeBERTA: 15.1%), while mDeBERTa scores the highest intent detection accuracy (73.5%).

For the remaining experiments, we use mDeBERTa as we expect the results of a multilingual model to be more generalizable when applied to other languages/dialects than the ones in our study.

6.2 Multi-Task Learning and Intermediate-Task Training

Both the choice of joint or sequential setup and the choice of auxiliary tasks influence the results (Table 1). Generally, the auxiliary tasks are more helpful for slot filling than for intent classification. This might be due to them, like slot filling, being on a token level. We could not include any sentence-level content classification tasks, for lack of datasets (cf. Blaschke et al., 2023).

Except for the MTL model with all auxiliary tasks, all settings that improve slot filling also help with intent classification, and vice versa.

Joint multi-task vs. intermediate-task training The **intermediate-task** setups (i.e., SID as a separate, last task) tend to beat the baseline in terms of both intent detection and slot filling, with gains of between 0.3 and 5.1 percentage points (pp.) for intent detection and between 3.8 and 9.0 for slot filling. The only exception is MLM→SID (–1.8 pp. for intents, –5.8 pp. for slots).[5] We assume that sep-

[5] In the set-ups where the model is first exclusively fine-tuned on MLM, the perplexity on the MLM development set is much higher than otherwise (Table 8 in Appendix §E),

arately fine-tuning the model on SID works well as the SID-related model weights cannot afterwards be modified by other tasks.

The **joint multi-task** setups (where SID is trained simultaneously with the other tasks), however, show less clear trends. Some task combinations have a large negative impact on intent classification (e.g., –24.6 pp. for UD×SID; –15.6 pp. when jointly fine-tuning on all tasks), while others have positive effects (e.g., +4.3 pp. for MLM×NER×SID). The effect on slot filling is much more positive, with performance differences ranging from –3.2 to +9.5 percentage points. Here, performance appears to depend more on the choice of auxiliary task:

Auxiliary task choice The **UD** tasks help when they are included as intermediary tasks, but lower the performance in nearly all joint MTL settings. This is somewhat similar to the results by van der Goot et al. (2021a), who found MTL with target-language UD tasks to mostly lower the intent classification performance but to have a mixed impact on slot filling.

Including **NER** as an auxiliary task is almost always beneficial for slot filling (and otherwise only has a small impact: –0.6 pp. for UD×NER×SID). We hypothesize that this is due to the high similarity between the two tasks (cf. Louvan and Magnini, 2020). It also has a positive effect on the intent classification performance, except in joint setups with UD and SID.

On its own, **MLM** has a negative effect on both slot filling and intent classification, regardless of whether it is included as a joint or intermediate-task. When it is, however, used together with other auxiliary tasks, it always improves the slot filling performance and nearly always helps the intent classification performance. These findings are somewhat different from the ones by van der Goot et al. (2021a), where joint MTL with target-language MLM improves slot filling performance and has mixed effects on intent classification. It is possible that the MLM dataset in our study is too small to meaningfully serve as data for continued pre-training, and that including more data would have made MLM a more beneficial task.

i.e., the auxiliary task was not learned properly. A possible explanation is that the standard hyperparameters might not have been optimal for MLM, and that the different model parameter updates in a multi-task learning context mitigated this somewhat.

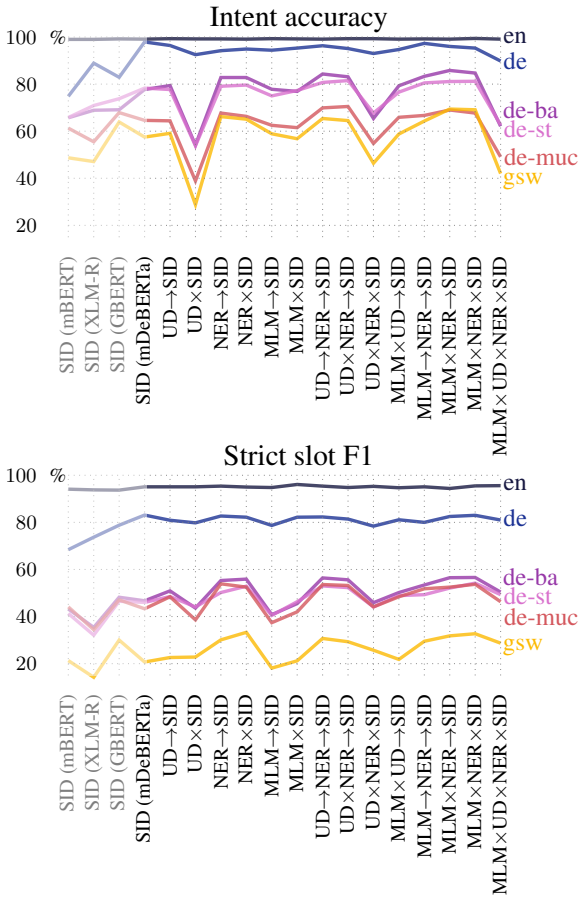

Figure 4: **Intent (top) and slot (bottom) scores show similar patterns across experimental set-ups for the test varieties.** The scores are averaged across three random seeds (more details are in Appendix D). The pale sections to the left show the scores of baseline models with different PLMs. We use lines despite the categorical nature of the x-axis to make the plots easier to compare.

6.3 Performance Differences Across Languages

While we previously focused on averages over the three Bavarian dialect datasets, we now compare the performance differences between them, and also analyze the test scores on related languages (Figure 4). The detailed prediction scores are in Appendix D, and we summarize the trends below.

Bavarian dialects While the scores differ across dialects, the trends across experimental setups are the same: A setup that is beneficial or damaging for the performance on one dialect has a similar effect on the others. The performance gaps for the multi-task and sequential settings are similar in scale to the gaps of the corresponding baseline.

The predictions on the Munich Bavarian (de-muc) test set tend be be worse than for the

Upper Bavarian (de-ba) and South Tyrolean (de-st) datasets. This is especially pronounced for the intent classification results (Figure 4, top). There, the results on de-ba and de-st are very similar, but the scores on de-muc are between 1.2 and 17.0 pp. lower than those on de-ba. The slot filling performance is more consistent across dialects (Figure 4, bottom), with score differences of 0.0–5.5 pp. between dialect pairs. Nevertheless, the results on de-ba tend to be slightly better than for the other dialects.

We discuss differences between the Bavarian test sets in §6.4.

Swiss German We additionally consider the performance on Bernese Swiss German, which, like the Bavarian dialects, belongs to the Upper German dialect group. Performance on Swiss German is always worse than on the Bavarian dialects – also for the baseline models that were not fine-tuned on Bavarian auxiliary tasks. This is in line with other SID systems evaluated on the gsw data (Aepli et al., 2023) and might be due to the translation being more dissimilar to Standard German than the Bavarian ones (Appendix A). However, the trends for Swiss German are similar as for the Bavarian dialects: Setups that improve or lower SID performance for Bavarian also do so for Swiss German, despite only involving Bavarian auxiliary data.

Standard German We analyze the performance on Standard German, which is part of mDeBERTa's pretraining dataset. Performance on Standard German is consistently better than on the Bavarian dialects (intent detection accuracy remains at $\geq$ 89.8%, slot filling F1 at $\geq$ 78.7%). Bavarian auxiliary tasks incur performance losses on the Standard German test data across all settings, but the settings that harm performance on Bavarian also have the most deteriorating effect on the predictions for German.

English Lastly, we turn to English – the fine-tuning language. The scores are barely affected by the auxiliary tasks: Intent detection accuracy remains at $\geq$ 99.1% (the same as for the baseline) and slot filling F1 scores at $\geq$ 94.4% (–0.7 pp.).

6.4 Differences Between Bavarian Translations

The test sets reflect differences between Bavarian dialects (§3) and translation choices. Table 2 shows translations of the English test sentence "Delete all alarms", which exhibit both spelling variation ("alarms" rendered as *wecka, Wegga, Wecker*) and different word choices (*streich* "remove", *lösch* "delete", and *tua ... weck* "do ... away").

DE-MUC

streich *remove*.IMP	olle *all*	wecka *alarms*	[intent]
O	B–ref.	O	alarm/cancel_alarm
B–entity _name ✗	I–rem./ todo ✗	O ✓	AddToPlaylist ✗

DE-BA

Lösch *delete*.IMP	olle *all*	Wegga *alarms*	[intent]
O	B–re.	O	alarm/cancel_alarm
O ✓	B–ref. ✓	O ✓	alarm/cancel_alarm ✓

DE-ST

tua *do*.IMP	olle *all*	Wecker *alarms*	weck *away*	[intent]
O	B–ref.	O	O	alarm/cancel_alarm
O ✓	O ✗	O ✓	O ✓	alarm/set_alarm ✗

Table 2: **Translations of "Delete all alarms" into Bavarian dialects with gold-standard and (correctly ✓ or incorrectly ✗) predicted annotations.** The predictions are by the overall best-performing model, MLM×NER→SID, with the same random seed. Abbreviated slots: ref. = reference, rem. = reminder.

Although there is very little morphosyntactic variation between Bavarian dialects, some of the translations exhibit different morphosyntactic structures that reflect different translation choices. Table 10 in Appendix G provides an example.

Even small differences between translations can affect the predictions of a SID model. In both examples, all three translations receive different slot and intent labels by the best-performing model in our experiments – even though the first two translations in Table 2 have an identical structure to the English sentence, which is annotated correctly.

One possible reason for this is that the Munich translation is mostly lower-cased, unlike the other Bavarian translations. This likely further decreases the subword token overlap with German cognates that might be in the PLM's pretraining data.

6.5 Additional Bavarian Test Sets

To investigate the robustness of our findings not only across dialects, but also across different datasets from the same area (Upper Bavaria; de-ba), we use the additional datasets mentioned at the end of §4. We evaluate the baseline model, the best-performing model (MLM×NER→SID), and its

MTL counterpart (MLM×NER×SID), which also performs well on the xSID data (Figures 3 and 4).

All three models perform best on the xSID data (intent accuracy: 77.7%, slot F1: 46.7%) and worst on the MASSIVE translations (intents: 55.2%, slots: 22.1%), with the naturalistic data in between (intents: 60.8%, slots: 31.7%). The detailed scores are in Appendix F (Table 9). The models that were also trained on auxiliary data nearly always improve over the baseline. The overall best-performing model incurs improvements of 6.7–7.9 pp. for intent classification and 9.7–9.9 pp. for slot filling on the additional test sets. Nevertheless, the magnitudes of the performance gains for each model are slightly different compared to the xSID data. Thus, while well-performing SID systems are also useful for data from other distributions, the performance patterns are not identical.

7 Conclusion

In all of our cross-lingual SID experiments, the performance patterns are similar across dialects, but the actual scores differ. To allow future research on this kind of variation, we release a new evaluation dataset (de-muc). In our experiments, intermediate-task training tends to produce better results than joint multi-task learning. Additionally, our Bavarian auxiliary tasks (POS tagging and dependency parsing, NER, MLM) were more beneficial for slot filling than intent classification, with NER being the overall most helpful auxiliary task.

Acknowledgments

We thank Rob van der Goot for useful discussions regarding MaChAmp, Siyao Peng for early discussions of the topic, Ryan Soh-Eun Shim for giving feedback on an early version of the draft, and the anonymous reviewers for their comments.

This research is supported by European Research Council (ERC) Consolidator Grant DIALECT 101043235.

Limitations

Data The dialect tags should not be taken to reflect all dialect speakers from the respective regions, nor necessarily the most traditional forms of these dialects. That is, the new de-muc development/test set only reflects the language of one young Munich Bavarian speaker (see also §B.11).

Tasks Due to lack of data, we could not conduct any experiments with sentence-level auxiliary tasks, and we also could not compare our results to settings with German or even Bavarian SID training data.

We include MLM as one of our auxiliary tasks since it is a common pre-training objective, albeit not the one used for mDeBERTa v.3 (He et al., 2021a), which instead uses replaced token detection (RTD; Clark et al., 2020). We use MLM as it is supported by MaChAmp, and selecting a (separate) MLM generator model for RTD would have introduced additional task-specific parameters.

PLMs In the paper by van der Goot et al. (2021a), the impact of the auxiliary tasks differs for two PLMs. Due to computational constraints, we only carried out the (non-baseline) experiments with a single PLM and did not evaluate how robust the results are across PLMs.

Implementation We decode the slot predictions with a simple softmax layer. This might lead to lower slot filling results than decoding the output with conditional random fields to enforce consistent BIO sequences (van der Goot et al., 2021a,b). We do not assume that changing the output decoder would lead to different trends regarding the effects of MTL and intermediate-task training.

We use MaChAmp's default settings, including the maximum number of epochs (20) to keep feasible computation times. In many experiments, the optimal number of epochs was 20 or close to 20. It is possible that we could have reached better results with a larger number of epochs. Training the model for longer might have been especially crucial for MLM. We hypothesize that this might have increased both the intermediate MLM and the final SID performance of the MLM→SID model (§6.2).

We also use the default settings for all tasks, including MLM. This leads to the MLM data being split across epochs, leaving only a small portion (70 sentences) being used per epoch. Disabling this split might have lead to better or more consistent MLM results.

References

Khadige Abboud and Gokmen Oz. 2024. Towards equitable natural language understanding systems for dialectal cohorts: Debiasing training data. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources*

and Evaluation (LREC-COLING 2024), pages 16487–16499, Torino, Italia. ELRA and ICCL.

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 *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pages 251–261, Dubrovnik, Croatia. Association for Computational Linguistics.

Ekaterina Artemova, Verena Blaschke, and Barbara Plank. 2024. Exploring the robustness of task-oriented dialogue systems for colloquial German varieties. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 445–468, St. Julian's, Malta. Association for Computational Linguistics.

Ekaterina Artemova and Barbara Plank. 2023. Low-resource bilingual dialect lexicon induction with large language models. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 371–385, Tórshavn, Faroe Islands. University of Tartu Library.

Josef Bayer. 1993. *Zum* in Bavarian and scrambling. In Werner Abraham and Josef Bayer, editors, *Dialektsyntax*. Westdeutscher Verlag.

Josef Bayer and Ellen Brandner. 2004. Klitisiertes *zu* im Bairischen und Alemannischen. In *Morphologie und Syntax deutscher Dialekte und Historische Dialektologie des Deutschen: Beiträge zum 1. Kongress der Internationalen Gesellschaft für Dialektologie des Deutschen*.

Emily M. Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Verena Blaschke, Barbara Kovačić, Siyao Peng, Hinrich Schütze, and Barbara Plank. 2024a. MaiBaam: A multi-dialectal Bavarian Universal Dependency treebank. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10921–10938, Torino, Italia. ELRA and ICCL.

Verena Blaschke, Christoph Purschke, Hinrich Schuetze, and Barbara Plank. 2024b. What do dialect speakers want? a survey of attitudes towards language technology for German dialects. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 823–841, Bangkok, Thailand. Association for Computational Linguistics.

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.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

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.

Hossam Elkordi, Ahmed Sakr, Marwan Torki, and Nagwa El-Makky. 2024. AlexuNLP24 at AraFinNLP2024: Multi-dialect Arabic intent detection with contrastive learning in banking domain. In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 415–421, Bangkok, Thailand. Association for Computational Linguistics.

Murhaf Fares and Samia Touileb. 2024. BabelBot at AraFinNLP2024: Fine-tuning t5 for multi-dialect

intent detection with synthetic data and model ensembling. In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 433–440, Bangkok, Thailand. Association for Computational Linguistics.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2023. MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4277–4302, Toronto, Canada. Association for Computational Linguistics.

Daniela Gerz, Pei-Hao Su, Razvan Kusztos, Avishek Mondal, Michał Lis, Eshan Singhal, Nikola Mrkšić, Tsung-Hsien Wen, and Ivan Vulić. 2021. Multilingual and cross-lingual intent detection from spoken data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7468–7475, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mutian He and Philip Garner. 2023. The interpreter understands your meaning: End-to-end spoken language understanding aided by speech translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4408–4423, Singapore. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*.

Alkis Koudounas, Moreno La Quatra, Lorenzo Vaiani, Luca Colomba, Giuseppe Attanasio, Eliana Pastor, Luca Cagliero, and Elena Baralis. 2023. ITALIC: An Italian intent classification dataset. In *INTERSPEECH 2023*, pages 2153–2157.

Vladimir Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710. [Russian original (1965) in Doklady Akademii Nauk SSSR, 163(4):845—848.].

Samuel Louvan and Bernardo Magnini. 2020. Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 480–496, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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.

Sanad Malaysha, Mo El-Haj, Saad Ezzini, Mohammed Khalilia, Mustafa Jarrar, Sultan Almujaiwel, Ismail Berrada, and Houda Bouamor. 2024. AraFinNLP 2024: The first Arabic financial NLP shared task. In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 393–402, Bangkok, Thailand. Association for Computational Linguistics.

Héctor Martínez Alonso and Barbara Plank. 2017. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 44–53, Valencia, Spain. Association for Computational Linguistics.

Ludwig Merkle. 1993. *Bairische Grammatik*, 5th edition. Heinrich Hugendubel Verlag, Munich.

Syrielle Montariol, Arij Riabi, and Djamé Seddah. 2022. Multilingual auxiliary tasks training: Bridging the gap between languages for zero-shot transfer of hate speech detection models. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 347–363, Online only. Association for Computational Linguistics.

Alberto Muñoz-Ortiz, Verena Blaschke, and Barbara Plank. 2025. Evaluating pixel language models on non-standardized languages. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING 2025)*, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Vishakh Padmakumar, Leonard Lausen, Miguel Ballesteros, Sheng Zha, He He, and George Karypis. 2022. Exploring the role of task transferability in large-scale multi-task learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2542–2550, Seattle, United States. Association for Computational Linguistics.

Siyao Peng, Zihang Sun, Huangyan Shan, Marie Kolm, Verena Blaschke, Ekaterina Artemova, and Barbara Plank. 2024. Sebastian, basti, wastl?! recognizing named entities in Bavarian dialectal data. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14478–14493, Torino, Italia. ELRA and ICCL.

Jason Phang, Iacer Calixto, Phu Mon Htut, Yada Pruksachatkun, Haokun Liu, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. English intermediate-task training improves zero-shot cross-lingual transfer too. In *Proceedings of the 1st Conference of the*

Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 557–575, Suzhou, China. Association for Computational Linguistics.

Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. 2021. What to pre-train on? Efficient intermediate task selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10585–10605, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247, Online. Association for Computational Linguistics.

Shangeth Rajaa, Swaraj Dalmia, and Kumarmanas Nethil. 2022. Skit-S2I: An Indian accented speech to intent dataset. *Preprint*, arXiv:2212.13015.

Asmaa Ramadan, Manar Amr, Marwan Torki, and Nagwa El-Makky. 2024. MA at AraFinNLP2024: BERT-based ensemble for cross-dialectal Arabic intent detection. In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 441–445, Bangkok, Thailand. Association for Computational Linguistics.

Anthony R. Rowley. 2023. *Boarisch – Boirisch – Bairisch: Eine Sprachgeschichte*. Verlag Friedrich Pustet, Regensburg.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *Preprint*, arXiv:1706.05098.

Louvan Samuel, Silvia Casola, and Bernardo Magnini. 2022. Investigating continued pretraining for zero-shot cross-lingual spoken language understanding. In *Proceedings of the Eighth Italian Conference on Computational Linguistics CliC-It 2021*. Accademia University Press.

Fynn Schröder and Chris Biemann. 2020. Estimating the influence of auxiliary tasks for multi-task learning of sequence tagging tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2971–2985, Online. Association for Computational Linguistics.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019a. 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.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019b. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. 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 1599–1613, Minneapolis, Minnesota. Association for Computational Linguistics.

Aarohi Srivastava and David Chiang. 2023. Fine-tuning BERT with character-level noise for zero-shot transfer to dialects and closely-related languages. In *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pages 152–162, Dubrovnik, Croatia. Association for Computational Linguistics.

Shyam Upadhyay, Manaal Faruqui, Gokhan Tür, Hakkani-Tür Dilek, and Larry Heck. 2018. (almost) zero-shot cross-lingual spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6034–6038.

Rob van der Goot, Ibrahim Sharaf, Aizhan Imankulova, Ahmet Üstün, Marija Stepanović, Alan Ramponi, Siti Oryza Khairunnisa, Mamoru Komachi, and Barbara Plank. 2021a. 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*, pages 2479–2497, Online. Association for Computational Linguistics.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021b. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics.

Jixuan Wang, Kai Wei, Martin Radfar, Weiwei Zhang, and Clement Chung. 2021. Encoding syntactic knowledge in transformer encoder for intent detection and slot filling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13943–13951.

Oskar Weise. 1910. Die Stundenbezeichnungen in den deutschen Mundarten. *Zeitschrift für Deutsche Mundarten*, 5:260–264.

Helmut Weiß. 1998. *Syntax des Bairischen*. Max Niemeyer Verlag.

Orion Weller, Kevin Seppi, and Matt Gardner. 2022. When to use multi-task learning vs intermediate fine-tuning for pre-trained encoder transfer learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2:*

Short Papers), pages 272–282, Dublin, Ireland. Association for Computational Linguistics.

Peter Wiesinger. 1983. Die Einteilung der deutschen Dialekte. In Werner Besch, Ulrich Knoop, Wolfgang Putschke, and Herbert E. Wiegand, editors, *Ergebnisse dialektologischer Beschreibungen: Areale Bereiche deutscher Dialekte im Überblick*, pages 807–960. De Gruyter Mouton, Berlin, Boston.

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 (LREC-COLING 2024)*, pages 14898–14915, Torino, Italia. ELRA and ICCL.

Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual NLU. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5052–5063, Online. Association for Computational Linguistics.

Marcos Zampieri, Preslav Nakov, and Yves Scherrer. 2020. Natural language processing for similar languages, varieties, and dialects: A survey. *Natural Language Engineering*, 26(6):595–612.

A Dataset Distances

We compare how similar the translations are to each other. For each pair of sentence translations, we calculate the word-level Levenshtein (1966) edit distance. We also select all words tagged as the same slot type (ignoring the `B` or `I` prefixes) and join them with blank spaces. For corresponding pair of slot values, we calculate the character-level edit distance. We normalize each distance by dividing it by the length of the longer phrase, and we convert it into a similarity score by subtracting it from 1.

For both similarity levels (sentences and slots) and regardless of whether we consider casing differences, the two Central Bavarian translations (de-ba, de-muc) are more similar to each other than any of the other pairs (Table 3). The Bavarian and Standard German translations are closer to each other than the Swiss German translation.

Slot similarity (chars), case sensitive

	de	de-ba	de-muc	de-st	gsw
en	0.51	0.51	0.55	0.48	0.42
de		0.69	0.66	0.73	0.58
de-ba			0.77	0.68	0.56
de-muc				0.67	0.51
de-st					0.55

Slot similarity (chars), case insensitive

	de	de-ba	de-muc	de-st	gsw
en	0.53	0.53	0.55	0.51	0.45
de		0.70	0.70	0.74	0.59
de-ba			0.81	0.69	0.58
de-muc				0.70	0.54
de-st					0.55

Sent similarity (words), case sensitive

	de	de-ba	de-muc	de-st	gsw
en	0.15	0.16	0.22	0.14	0.08
de		0.27	0.20	0.33	0.14
de-ba			0.45	0.29	0.13
de-muc				0.24	0.13
de-st					0.13

Sent similarity (words), case insensitive

	de	de-ba	de-muc	de-st	gsw
en	0.17	0.19	0.22	0.16	0.10
de		0.28	0.24	0.33	0.14
de-ba			0.50	0.30	0.13
de-muc				0.27	0.14
de-st					0.13

Table 3: **Mean similarities between slots or sentences corresponding to each other.** The similarities are calculates as 1 minus the normalized Levenshtein distance.

B Data Statement

B.1 Header

- *Dataset Title:* xSID de-muc
- *Dataset Curator(s):* Xaver Maria Krückl, Verena Blaschke, Barbara Plank
- *Dataset Version:* 1.0 (expected to be part of xSID 0.7)
- *Dataset Citation:* Please cite this paper when using this dataset.
- *Data Statement Authors:* Xaver Maria Krückl, Verena Blaschke, Barbara Plank
- *Data Statement Version:* 1.0
- *Data Statement Citation and DOI:* Please cite this paper when referring to the data statement.
- *Links to versions of this data statement in other languages:* —

B.2 Executive Summary

xSID de-muc is a manually annotated (translated) extension of the English xSID development and train set (van der Goot et al., 2021a) into the Bavarian dialect spoken in Munich. The development set contains 300 translated samples and the test set 500. The intents were taken over from the English gold examples whereas the slots were annotated by the translator. The translations were made over several weeks.

B.3 Curation Rationale

The purpose of xSID de-muc is to provide further dialectal development and test data in addition to other Bavarian translations. We hope to extend our research on dialectal SID through our data.

B.4 Documentation for Source Datasets

The xSID de-muc development and test set are based on the respective English sets from xSID (van der Goot et al., 2021a; CC BY-SA 4.0), which in turn are derived in equal parts from two larger datasets, the Snips (Coucke et al., 2018; CC0 1.0 Universal) and Facebook (Schuster et al., 2019a; CC-BY-SA license) datasets.

B.5 Language Varieties

xSID de-muc contains data in Munich Bavarian (a Central Bavarian dialect), as spoken by a young speaker.

B.6 Language User Demographic

The original data were created by crowd workers whose demographics are not known. For the translator, see *Annotator Demographic*.

B.7 Annotator Demographic

The translator and annotator is a native speaker of German and Munich Bavarian in his mid-twenties. He annotated the data while finishing his Master's degree in Computational Linguistics and is one of the authors of this paper.

B.8 Linguistic Situation and Text Characteristics

xSID consists of random samples from the English Snips (Coucke et al., 2018) and Facebook (Schuster et al., 2019a) datasets, which are compiled from utterances to be used for training digital assistants. Both datasets were mainly crowd-sourced; annotations were validated.

B.9 Preprocessing and Data Formatting

We directly worked with xSID's English sentences and did not apply any further preprocessing steps. Like the rest of xSID, the data set is in the CONLL format.

B.10 Capture Quality

Some sentences contain grammatical errors or typos in the original datasets. Following xSID's translation guidelines, we retained such errors in the de-muc translations.

B.11 Limitations

The data set is a translation, which probably differs from the way speakers express themselves when not prompted to translate (Winkler et al., 2024) or in fluent conversation.

It reflects the language use of a single speaker. It does not represent the most traditional form of Munich Bavarian. Additionally, other speakers might prefer other spellings (since Bavaria has no established orthography).

B.12 Metadata

- *Annotation Guidelines:* Appendices F and G of van der Goot et al. (2021a)
- *Annotation Process:* — (see this paper)
- *Dataset Quality Metrics:* —

B.13 Disclosures and Ethical Review

There are no conflicts of interest. This research is supported by European Research Council (ERC) Consolidator Grant DIALECT 101043235.

B.14 Distribution

The de-muc split will be included in xSID under the same license, accessible via `https://github.com/mainlp/xsid`.

B.15 Maintenance

Errors can be reported via GitHub issues or emailing us. Updates to the dataset (and the release history) will be available in the repository.

B.16 Other

—

B.17 Glossary

—

About this document

A data statement is a characterization of a dataset that provides context to allow developers and users to better understand how experimental results might generalize, how software might be appropriately deployed, and what biases might be reflected in systems built on the software.

This data statement was written based on the template for the Data Statements Version 3 Schema. The template was prepared by Angelina McMillan-Major and Emily M. Bender and can be found at `http://techpolicylab.uw.edu/data-statements`.

C Baseline Systems

Table 4 shows the results of our baseline systems (no auxiliary tasks) and the baseline systems by van der Goot et al. (2021a) on all languages that were in the original xSID release. Note that we use XLM-R while van der Goot et al. (2021a) use XLM-15.

D Detailed Results

We include tables with detailed results for the Bavarian dialects, in addition to results for Swiss German, German, and English. Table 5 shows the intent classification scores, Table 6 the slot detection scores, and Table 7 for fully correct classifications (slots and intents).

E Auxiliary Task Scores

Table 8 shows the scores on the development sets of the auxiliary tasks.

F Additional Bavarian Test Sets

Table 9 shows results on the de-ba dataset in addition to other data in the same dialect (or dialects spoken in the same region).

G Additional Examples

Table 10 provides another example for translation (and prediction) differences between the Bavarian dialects.

	ar	da	de	de-st	en	id	it	ja	kk	nl	sr	tr	zh
Intents (accuracy, in %)													
mBERT (vdG)	63.1	87.5	74.2	67.8	99.7	80.7	81.7	53.9	60.1	72.3	75.7	74.7	83.3
mBERT	67.9	84.8	74.8	65.8	99.0	76.0	76.3	55.5	56.9	69.9	75.7	71.3	84.8
XLM-15 (vdG)	65.5	56.3	78.5	61.3	99.7	36.4	48.0	39.1	29.9	45.4	41.4	67.3	78.8
XLM-R	78.1	95.3	88.9	70.9	99.0	95.3	80.5	54.5	75.8	84.3	82.7	94.1	96.0
GBERT	27.2	61.9	82.9	73.8	99.2	46.9	52.3	5.6	34.9	59.1	45.7	46.6	23.8
mDeBERTa	86.9	96.5	97.9	78.3	99.1	96.3	97.4	79.2	89.9	96.5	89.1	97.2	96.9
Slots (strict slot F1, in %)													
mBERT (vdG)	45.8	73.9	33.0	48.5	97.6	71.1	75.0	59.9	48.5	80.4	67.4	55.7	72.9
mBERT	52.4	70.3	68.4	41.3	94.1	63.8	69.9	39.4	32.2	70.1	55.0	32.9	48.0
XLM-15 (vdG)	49.1	26.3	33.3	39.4	97.0	14.9	27.3	33.4	10.9	30.9	15.9	45.5	57.6
XLM-R	62.3	80.9	73.7	32.1	93.8	76.6	75.6	51.0	45.2	82.2	63.9	52.9	66.8
GBERT	19.7	37.3	78.8	46.5	93.7	17.2	28.0	0.7	5.4	44.4	18.5	8.3	14.5
mDeBERTa	71.1	79.7	83.1	46.0	95.1	78.3	83.1	49.8	52.4	86.6	72.1	58.3	74.7
Fully correct (in %)													
mBERT	18.5	44.5	34.6	9.5	88.3	31.6	37.7	20.3	8.5	37.1	24.6	12.4	15.9
XLM-R	28.8	64.4	49.3	6.9	88.5	56.0	47.4	25.9	16.9	57.7	38.0	34.6	46.4
GBERT	4.9	12.4	53.7	14.7	87.5	1.9	4.5	0.9	1.9	12.8	3.9	2.3	3.3
mDeBERTa	44.1	61.7	66.1	15.9	90.0	58.8	63.1	40.4	24.6	72.7	46.3	35.6	57.7

Table 4: **Scores of our baselines on xSID's original test language selection.** We also include scores by van der Goot et al. (2021a) for comparison (= vdG). XLM-15 refers to xlm-mlm-tlm-xnli15-1024 (Conneau and Lample, 2019).

	de-muc	de-ba	de-st	gsw	de	en
SID (mBERT)	$61.3_{1.2}$	$65.7_{2.4}$	$65.8_{2.0}$	$48.7_{2.2}$	$74.8_{1.7}$	$99.0_{0.2}$
SID (XLM-R)	$55.5_{2.6}$	$68.9_{0.7}$	$70.9_{1.2}$	$47.1_{2.6}$	$88.9_{1.7}$	$99.0_{0.2}$
SID (GBERT)	$67.9_{2.7}$	$69.1_{0.5}$	$73.8_{1.0}$	$63.9_{1.5}$	$82.9_{1.3}$	$99.2_{0.0}$
SID (mDeBERTa)	$64.6_{3.1}$	$77.7_{0.7}$	$78.3_{0.8}$	$57.5_{2.7}$	$97.9_{0.4}$	$99.1_{0.1}$
UD→SID	$64.4_{4.0}$	$79.4_{2.3}$	$77.7_{2.4}$	$59.1_{4.6}$	$96.4_{1.0}$	$99.3_{0.2}$
UD×SID	$38.9_{3.2}$	$54.1_{2.8}$	$53.8_{2.5}$	$28.8_{1.4}$	$92.5_{1.7}$	$99.2_{0.2}$
NER→SID	$67.7_{0.5}$	$82.8_{3.5}$	$79.1_{1.3}$	$66.2_{3.2}$	$94.2_{0.7}$	$99.2_{0.2}$
NER×SID	$66.3_{1.6}$	$82.8_{1.8}$	$79.6_{1.0}$	$65.1_{1.1}$	$94.9_{1.2}$	$99.1_{0.1}$
MLM→SID	$62.5_{2.2}$	$77.8_{2.9}$	$75.0_{1.8}$	$58.9_{5.2}$	$94.4_{2.4}$	$99.3_{0.2}$
MLM×SID	$61.5_{2.1}$	$76.9_{2.1}$	$77.3_{0.5}$	$56.8_{3.4}$	$95.3_{1.3}$	$99.2_{0.2}$
UD→NER→SID	$69.9_{1.0}$	$84.3_{2.9}$	$80.7_{1.5}$	$65.4_{1.8}$	$96.3_{0.7}$	$99.1_{0.1}$
UD×NER→SID	$70.5_{1.8}$	$83.1_{1.7}$	$81.5_{1.2}$	$64.5_{3.6}$	$95.1_{2.7}$	$99.3_{0.2}$
UD×NER×SID	$54.8_{2.4}$	$65.4_{3.5}$	$67.7_{1.1}$	$46.4_{3.1}$	$93.0_{0.9}$	$99.3_{0.1}$
MLM×UD→SID	$65.9_{1.0}$	$79.3_{1.4}$	$76.6_{2.5}$	$58.8_{1.0}$	$94.7_{1.5}$	$99.1_{0.2}$
MLM→NER→SID	$66.7_{1.0}$	$83.3_{1.6}$	$80.5_{1.1}$	$64.3_{1.9}$	$97.3_{0.5}$	$99.2_{0.2}$
MLM×NER→SID	$69.0_{1.7}$	$85.8_{1.3}$	$81.1_{0.7}$	$69.4_{1.7}$	$96.0_{1.1}$	$99.1_{0.1}$
MLM×NER×SID	$67.7_{1.0}$	$84.7_{0.5}$	$81.2_{2.6}$	$69.1_{3.3}$	$95.3_{1.0}$	$99.4_{0.2}$
MLM×UD×NER×SID	$49.1_{5.8}$	$62.1_{4.6}$	$62.7_{3.6}$	$42.0_{8.3}$	$89.8_{0.9}$	$99.1_{0.2}$

Table 5: **Intent classification results in the three Bavarian dialects, Swiss German, German, and English.** We show mean scores (accuracy, in %) over three random seeds, with standard deviations in subscripts.

	de-muc	de-ba	de-st	gsw	de	en
SID (mBERT)	$44.1_{1.1}$	$43.2_{1.1}$	$41.3_{1.0}$	$21.3_{1.3}$	$68.4_{1.2}$	$94.1_{0.3}$
SID (XLM-R)	$34.4_{1.6}$	$35.4_{0.7}$	$32.1_{0.7}$	$14.2_{0.6}$	$73.7_{1.3}$	$93.8_{0.5}$
SID (GBERT)	$47.1_{1.3}$	$48.2_{0.7}$	$46.5_{2.6}$	$30.0_{0.9}$	$78.8_{0.8}$	$93.7_{0.4}$
SID (mDeBERTa)	$43.3_{0.9}$	$46.7_{2.0}$	$46.0_{1.0}$	$20.7_{2.5}$	$83.1_{0.7}$	$95.1_{0.2}$
UD→SID	$48.4_{2.5}$	$50.9_{0.2}$	$48.5_{2.2}$	$22.6_{0.3}$	$80.9_{1.5}$	$95.1_{0.1}$
UD×SID	$38.6_{2.9}$	$43.6_{4.2}$	$44.1_{4.0}$	$22.8_{3.4}$	$79.8_{1.2}$	$95.1_{0.2}$
NER→SID	$53.9_{0.5}$	$55.3_{2.3}$	$50.2_{1.4}$	$30.1_{1.4}$	$82.7_{0.6}$	$95.4_{0.3}$
NER×SID	$52.5_{1.8}$	$55.9_{1.1}$	$52.9_{0.9}$	$33.3_{0.1}$	$82.2_{1.1}$	$95.0_{0.2}$
MLM→SID	$37.4_{2.9}$	$40.8_{4.1}$	$40.5_{3.4}$	$18.1_{2.0}$	$78.7_{1.6}$	$94.8_{0.6}$
MLM×SID	$42.0_{1.4}$	$45.5_{2.0}$	$46.3_{1.6}$	$21.2_{1.8}$	$82.2_{0.7}$	$96.1_{0.2}$
UD→NER→SID	$53.6_{1.9}$	$56.4_{3.6}$	$53.0_{3.2}$	$30.7_{2.4}$	$82.3_{1.9}$	$95.4_{0.5}$
UD×NER→SID	$53.2_{1.6}$	$55.6_{1.7}$	$52.3_{0.5}$	$29.3_{0.4}$	$81.4_{1.3}$	$94.8_{0.7}$
UD×NER×SID	$44.1_{2.6}$	$45.9_{5.1}$	$44.1_{5.3}$	$25.7_{5.0}$	$78.4_{2.4}$	$95.3_{0.4}$
MLM×UD→SID	$48.4_{2.6}$	$50.2_{3.3}$	$48.9_{1.8}$	$21.8_{2.2}$	$81.1_{0.9}$	$94.7_{0.3}$
MLM→NER→SID	$51.8_{1.7}$	$53.4_{0.9}$	$49.3_{1.4}$	$29.5_{0.9}$	$80.0_{0.3}$	$95.1_{0.3}$
MLM×NER→SID	$52.5_{1.1}$	$56.5_{0.9}$	$52.1_{1.6}$	$31.8_{1.9}$	$82.5_{0.4}$	$94.4_{0.8}$
MLM×NER×SID	$53.7_{1.1}$	$56.6_{0.3}$	$54.2_{1.5}$	$32.7_{0.9}$	$83.0_{0.9}$	$95.5_{0.3}$
MLM×UD×NER×SID	$46.3_{1.2}$	$50.4_{2.3}$	$49.3_{1.1}$	$28.7_{0.8}$	$81.0_{0.7}$	$95.6_{0.4}$

Table 6: **Slots classification results in the three Bavarian dialects, Swiss German, German, and English.** We show mean scores (strict slot F1, in %) over three random seeds, with standard deviations in subscripts.

	de-muc	de-ba	de-st	gsw	de	en
SID (mBERT)	$11.0_{0.2}$	$13.4_{0.3}$	$9.5_{0.2}$	$3.0_{0.3}$	$34.6_{1.6}$	$88.3_{0.2}$
SID (XLM-R)	$6.3_{1.1}$	$11.3_{1.2}$	$6.9_{0.8}$	$1.6_{0.4}$	$49.3_{2.9}$	$88.5_{0.7}$
SID (GBERT)	$15.9_{0.5}$	$15.5_{1.2}$	$14.7_{2.3}$	$7.5_{0.9}$	$53.7_{3.0}$	$87.5_{0.6}$
SID (mDeBERTa)	$12.4_{2.0}$	$17.1_{1.3}$	$15.9_{0.9}$	$5.3_{1.1}$	$66.1_{1.1}$	$90.0_{0.4}$
UD→SID	$17.7_{1.5}$	$21.3_{0.4}$	$18.0_{2.0}$	$5.1_{0.7}$	$63.3_{1.7}$	$90.3_{0.4}$
UD×SID	$10.2_{0.7}$	$14.9_{2.9}$	$13.8_{1.3}$	$3.8_{1.0}$	$57.8_{2.9}$	$90.5_{0.5}$
NER→SID	$21.6_{1.1}$	$24.9_{3.1}$	$19.6_{2.0}$	$7.9_{1.1}$	$64.7_{1.2}$	$91.0_{0.3}$
NER×SID	$18.5_{2.4}$	$25.6_{1.0}$	$19.6_{1.4}$	$10.1_{0.3}$	$63.5_{0.9}$	$90.3_{0.5}$
MLM→SID	$9.0_{2.3}$	$14.9_{2.4}$	$12.3_{2.6}$	$3.9_{0.9}$	$58.3_{3.1}$	$90.1_{0.9}$
MLM×SID	$11.9_{1.5}$	$16.4_{2.7}$	$14.6_{1.3}$	$3.8_{0.0}$	$63.2_{1.5}$	$91.9_{0.3}$
UD→NER→SID	$21.5_{0.2}$	$24.9_{3.7}$	$21.5_{3.2}$	$7.4_{0.6}$	$65.1_{3.5}$	$90.7_{1.0}$
UD×NER→SID	$21.1_{1.6}$	$25.5_{1.9}$	$20.4_{1.6}$	$7.3_{0.3}$	$63.1_{1.4}$	$89.9_{0.8}$
UD×NER×SID	$14.0_{1.2}$	$16.7_{2.6}$	$14.7_{2.3}$	$5.9_{2.3}$	$57.4_{3.2}$	$90.7_{0.5}$
MLM×UD→SID	$18.2_{1.7}$	$21.2_{1.3}$	$18.5_{1.8}$	$5.3_{0.1}$	$61.7_{2.0}$	$89.5_{1.0}$
MLM→NER→SID	$19.1_{1.1}$	$23.5_{1.1}$	$20.7_{0.6}$	$7.0_{1.3}$	$63.4_{0.7}$	$90.3_{0.5}$
MLM×NER→SID	$20.5_{0.2}$	$25.7_{2.4}$	$22.6_{1.6}$	$9.2_{0.5}$	$65.5_{0.7}$	$89.5_{1.1}$
MLM×NER×SID	$20.1_{0.6}$	$25.6_{2.0}$	$21.3_{1.2}$	$10.3_{1.4}$	$64.9_{1.6}$	$91.2_{0.7}$
MLM×UD×NER×SID	$15.1_{1.3}$	$19.1_{2.3}$	$16.7_{1.6}$	$7.4_{1.1}$	$58.9_{1.5}$	$91.1_{0.5}$

Table 7: **Proportions of fully correctly classified sentences (slots and intents) in the three Bavarian dialects, Swiss German, German, and English.** We show mean scores (in %) over three random seeds, with standard deviations in subscripts.

	Dev scores				Test scores	
	LAS ↑	POS ↑	NER ↑	PPL ↓	Intents ↑	Slots ↑
SID (mDeBERTa)					$73.5_{6.6}$	$45.3_{2.0}$
UD→SID	$74.5_{0.6}$	$84.8_{0.3}$			$73.8_{7.3}$	$49.3_{2.2}$
UD×SID	$58.8_{9.8}$	$84.3_{0.7}$			$48.9_{7.6}$	$42.1_{4.5}$
NER→SID			$73.5_{1.3}$		$76.6_{6.8}$	$53.1_{2.7}$
NER×SID			$65.0_{1.0}$		$76.2_{7.3}$	$53.8_{2.0}$
MLM→SID				$436.4_{22.2}$	$71.8_{7.1}$	$39.6_{3.8}$
MLM×SID				$5.8_{0.3}$	$71.9_{7.6}$	$44.6_{2.5}$
UD→NER→SID	$75.2_{0.7}$	$85.6_{0.9}$	$72.6_{0.4}$		$78.3_{6.4}$	$54.3_{3.3}$
UD×NER→SID	$77.4_{9.1}$	$90.0_{0.1}$	$71.4_{1.0}$		$78.4_{5.8}$	$53.7_{1.9}$
UD×NER×SID	$67.2_{9.4}$	$86.4_{0.2}$	$63.9_{0.7}$		$62.6_{6.2}$	$44.7_{4.6}$
MLM×UD→SID	$75.5_{4.0}$	$86.1_{0.7}$		$44.8_{1.3}$	$74.0_{6.0}$	$49.2_{2.7}$
MLM→NER→SID			$70.1_{2.6}$	$436.4_{22.2}$	$76.8_{7.4}$	$51.5_{2.2}$
MLM×NER→SID			$72.9_{0.6}$	$7.0_{1.8}$	$78.6_{7.2}$	$53.7_{2.3}$
MLM×NER×SID			$66.3_{0.3}$	$5.7_{0.4}$	$77.9_{7.5}$	$54.8_{1.7}$
MLM×UD×NER×SID	$72.8_{1.8}$	$86.6_{0.7}$	$64.0_{0.6}$	$5.5_{0.2}$	$58.0_{7.9}$	$48.7_{2.4}$

Table 8: **Development set scores for the auxiliary tasks** (LAS = labelled attachment score; POS = POS tagging accuracy; NER = NER span F1; PPL = masked token perplexity). For context, we also show the intent accuracy and slot-filling span F1 score on the Bavarian test sets. All scores are averaged over three runs, the SID scores are additionally averaged over the three Bavarian test sets. Subscript numbers are standard deviations. Darker background colours indicate better results for the auxiliary task scores. For the SID results, green cell backgrounds indicate better results than the baseline, and red worse results.

	Intents (acc., in %)			Slots (span F1, in %)			Fully correct (in %)		
	de-ba	nat.	MAS.	de-ba	nat.	MAS.	de-ba	nat.	MAS.
SID (mDeBERTa)	$77.7_{0.7}$	$60.8_{1.4}$	$55.2_{3.5}$	$46.7_{2.0}$	$31.7_{2.3}$	$22.1_{1.4}$	$17.1_{1.3}$	$12.9_{1.6}$	$6.7_{1.0}$
MLM×NER×SID	$84.7_{0.5}$	$61.0_{3.9}$	$53.8_{2.5}$	$56.6_{0.3}$	$42.3_{2.1}$	$30.3_{0.9}$	$25.6_{2.0}$	$20.3_{1.3}$	$10.6_{0.8}$
MLM×NER→SID	$85.8_{1.3}$	$67.5_{1.3}$	$60.1_{1.2}$	$56.5_{0.9}$	$41.4_{2.5}$	$32.0_{1.4}$	$25.7_{2.4}$	$20.2_{1.0}$	$12.4_{0.4}$

Table 9: **Performances on different data sets with dialects from Upper Bavaria:** xSID (de-ba), naturalistic data (nat.), and a translated subset of MASSIVE (MAS.). The scores are averaged across three random seeds, with standard deviations in subscripts.

DE-MUC (intent: reminder/set_reminder, predicted: weather/find ✗)

Erinnad	mi	dass	i	morgn	papia	tiacha	im	lodn	hoi
Remind	*me*	*that*	*I*	*tomorrow*	*paper*	*towels*	*in.the*	*store*	*fetch.*1SG
O	O	O	O	B–datet.	B–rem./ todo	I–rem./ todo	I–rem./ todo	I–rem./ todo	I–rem./ todo
O ✓	O ✓	O ✓	O ✓	B–datet. ✓	B–rem./ todo ✓	O ✗	O ✗	O ✗	O ✗

DE-BA (intent: reminder/set_reminder, predicted: reminder/set_reminder ✓)

Erinner	mi	moang	Papiertaschentücher	im	Ladn	zum	hoin
Remind	*me*	*tomorrow*	*paper towels*	*in.the*	*store*	PART+DET	*fetch.*INF *(nominalized)*
O	O	B-datet.	B–rem./todo	I–rem./ todo	I–rem./ todo	I–rem./ todo	I–rem./todo
O ✓	O ✓	O ✗	B–rem./todo ✓	I–rem./ todo ✓	I–rem./ todo ✓	I–rem./ todo ✓	I–rem./todo ✓

DE-ST (intent: reminder/set_reminder, predicted: reminder/set_reminder ✓)

Erinner	mi	morgn	in	Gscheft	a	Küchnrolle	zi	kafn
Remind	*me*	*tomorrow*	*in(.the)*	*store*	*a*	*kitchen roll*	*to*	*buy.*INF
O	O	B-datet.	B-rem./ todo	I–rem./ todo	I–rem./ todo	I–rem./ todo	I–rem./ todo	I–rem./todo
O ✓	O ✓	B-datet. ✓	O ✗	I–rem./ todo ✓	I–rem./ todo ✓	I–rem./ todo ✓	I–rem./ todo ✓	I–rem./todo ✓

Table 10: **Translations of "Remind me to get paper towels at the store tomorrow" into Bavarian dialects with gold-standard and (correctly ✓ or incorrectly ✗) predicted annotations.** Note the different syntactic structures for expressing the infinitive or subordinated phrase, the different translations used for "store" and "paper towels" (and the different order in which they are mentioned), and the spelling differences (e.g., for "tomorrow"). The predictions are by the overall best-performing model, MLM×NER→SID, with the same random seed. Abbreviated slots: datet. = datetime, rem. = reminder.