Asking the Right Questions: Exploiting Hidden Interactions in a Generative Framework for Multilingual, Multitask Classification

Sebastian-Antonio Toma, Camelia Lemnaru, Vlad-Andrei Negru and **Rodica Potolea**
Computer Science Department, Technical University of Cluj-Napoca, Romania
Toma.So.Sebastian@student.utcluj.ro,
{Camelia.Lemnaru, Vlad.Negru, Rodica.Potolea}@cs.utcluj.ro

Abstract

This study explores the potential of leveraging additional training data as instructional prompts for a generative model in a multilingual, multitask recipe classification problem. By incorporating different tasks as additional questions, derived from data available only during fine-tuning, we aim to improve the classification performance of a sequence-to-sequence model for all tasks and languages involved. Furthermore, we investigate the impact that prompt-engineering has on the additional questions during fine-tuning, uncovering its significant role in helping the model learn hidden interactions between tasks. The proposed method produces absolute improvements of 2.3%, 6.22%, and 10.7% respectively in weighted multilingual accuracy (on three targeted classification tasks). The most effective additional actions are the questions derived from supplementary data, while the size of the model and whether we perform in-domain pre-training do not improve the final performance significantly. Our findings also underline the importance of training data selection and questioning strategies, especially in underrepresented languages, where we obtained an absolute increase in accuracy of 34.8% in the few-shot setting and 30.33% in the 0-shot setting for an underrepresented language in a difficult main task, together with an increase from 0% to 97% in F1-score for the most underrepresented class.

1 Introduction

Text classification has become increasingly important for effectively analyzing vast amounts of textual data across different languages, in the context of the diverse and growing multilingual landscape of digital content (Li et al., 2021). It can also be applied in a Multi-Task Learning (MTL) setup, with the aim to improve the performance and efficiency of Natural Language Processing (NLP) models by simultaneously learning multiple, related tasks (Hupkes et al., 2023).

This paper focuses on the application of MTL techniques in the context of multilingual text classification, with the declared goal of leveraging the inherent relationships between different classification tasks to improve the accuracy and robustness of the model used. We employ a multilingual generative model as the backbone and focus on three classification tasks, where the labels represent cooking properties for oven recipes. In addition to the labels for the three target tasks, the multilingual data contains annotations related to other properties of the dishes, such as dish type, size, certain ingredients, or oven settings. The proposed framework effectively captures both the information shared between the three target tasks, but also capitalizes on the additional annotations (available only during fine-tuning), by introducing new tasks through the use of instructional prompts (or questions). Thus, the model is able to explicitly learn from the synergies between the new tasks and the target tasks during the fine-tuning phase.

The main contribution of the paper is to propose an instruction-driven, joint learning generative framework that helps the model extract hidden correlations for a better classification of recipes, especially for underrepresented languages and classes, in a highly imbalanced dataset. Unlike Wu et al. (2022), from which our approach is inspired, our study generates the content of these instructional prompts (or questions) from auxiliary annotated sparse data available only during training, the model being charged with understanding and predicting the respective answers.

Additionally, we:

- perform an ablation study on the selection of the most relevant annotations to use to generate instructions, also guided by their correlation with the target problems, obtaining an increase in the overall accuracy of 2.3% in the few-shot setting and 5.22% in 0-shot

and 15.56% 0-shot for an underrepresented language.

- explore the benefit of in-domain pre-training, which produced some improvements for some of the problems and languages, but no consistent behavior was observed.

- perform evaluations in several fine-tuning scenarios: 0-shot vs. few-shot evaluations, using underrepresented vs. well-represented languages only, and using various sizes of the backbone language model, obtaining increases in absolute weighted accuracy of 2.3%, 6.22%. and 10.7% per problem, and 34.8% few-shot and 30.33% 0-shot for a difficult problem in an underrepresented language.

Finally, we obtain an improvement in the F1-score for some of the lowest sampled classes, in the underrepresented languages, from 0% to 97%, proving classification on highly imbalanced datasets can benefit from our method.

2 Related work

The idea of jointly training a single model on multiple tasks to enable the sharing of knowledge and representations across tasks has been explored in various NLP applications, from intent detection and slot filling, to joint entity classification, relation classification, and co-reference clusters in scientific literature, or machine translation (Chen et al., 2021). Lăpuşan et al. (2022) apply such an approach on BERT (Devlin et al., 2019) and other variations (RoBERTa, Liu et al., 2019), to perform German Cooking Recipe Classification on four labels related to oven parameters. They obtained the best classification performance using the domain-adapted pre-trained language model on the recipe title concatenated with the instructions, in a joint training regime.

The advent of generative models has opened up a new range of possibilities for developing mechanisms that efficiently exploit the compositional capabilities of language models through prompting, either via fine-tuning, in-context learning or even augmenting them with reasoning skills and external tools (Mialon et al., 2023, Al-Negheimish et al., 2021). Liu et al. (2021) present a novel paradigm that shows how prompt-engineering can be used to discard the pre-train and fine-tune approach of Large Language Models and replace it with “pre-train, prompt, and predict” methodology, using pre-defined prompts to reveal the model prior knowledge. Wu et al. (2022) introduce a Unified Generative Framework (UGEN) to model the tasks as question-answering problems for joint multiple Intent Detection and Slot Filling. They use a template of five questions during training to extract relevant information from the context, such as keywords, that will help the model to better generalize during the evaluation phase, where only two out of five questions were used (without the auxiliary helping questions used during training). Chain of Thought Prompting (CoT, Wei et al., 2023) explores the emergence of complex reasoning capabilities in LLMs via prompting in a sequence of carefully selected demonstrations.

3 The data

Our target is to improve multi-task, multi-lingual classification behavior using additional data available only during training, by using the MTL paradigm. To accomplish this, we used a private dataset in the cooking domain having these additional annotations and some difficult main tasks that can benefit from this method. It consists of tens of thousands of online cooking recipes scrapped from web (including a wide range of **public** cooking websites), in 6 languages: English, German, Dutch, Italian, French, and Swedish. German and English are overrepresented in the labeled data, while Italian and Swedish are the most underrepresented. Each recipe has been annotated for three target tasks:

- Meat-centric (**M.**) - binary text label with "yes" or "no" classes, representing whether the main "focus" of the dish is (a loaf of) meat

- Surface aspect level (**S.l.**) - binary numeric label: classes "1" & "2", representing the cooked product’s aspect/color

- Dehydration level (**D.l.**) - label with 5 numeric classes: from "1" to "5", representing the moisture reduction degree.

Additionally, besides the main classification tasks of **M.**, **S.l.**, and **D.l.**, each recipe has:

- Title

- Instructions

- Oven settings

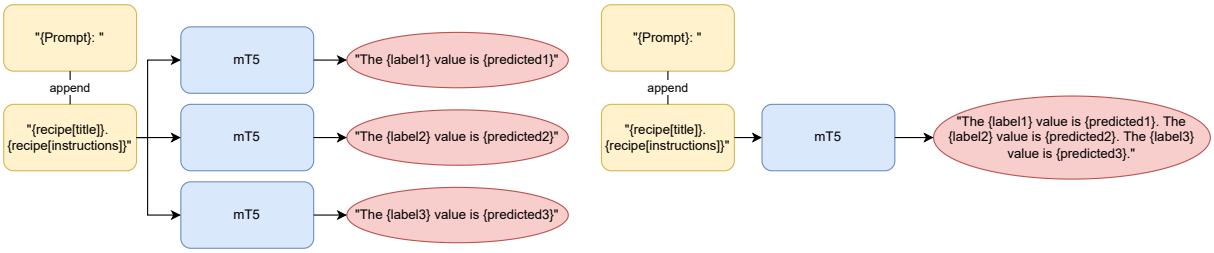

Figure 1: Architecture of the baseline model (left) vs. architecture of the joint model (right).

Initial setting	Final setting	Description
$a \vee b$	b	"OR" operands with missing/neutral values are discarded
$d \vee b$	b	
$c \vee b$	c	If both have missing/neutral values or both complete, take first

Table 1: Pre-processing procedure of multi-step recipes' oven settings, where a = *"T: 200°C, t: 25min, P: Not known"*, b = *"T: 100°C, t: 25min, P: Gas"*, c = *"T: 212°F, t: 25min, P: Grill"* and d = *"T: null, t: 25min, P: Gas"*.

- Some additional annotations related to the required cooking settings, dish type, certain ingredients, size, type, thickness, etc.

For each recipe, the default text used by the model as input consists of the recipe's **title** and **instructions** extracted from the HTML content, as in Lăpuşan et al. (2022).

4 Proposed Method

The selected **Baseline** model uses the default input (*title+instructions*) to predict only one of the three main tasks (**M.** or **S.l.** or **D.l.**), resulting in three baseline models, one for each.

Next, each incremental step is described together with the name used for the model and the part of the dataset being added.

4.1 Joint Learning

To better exploit the hidden correlations between the target problems, we first trained a **Joint** model, fine-tuned on all three tasks combined into one. The chosen order of generation is from the easiest task to the hardest task (as indicated by baseline, individual models: lower accuracy obtained for the task, means for us that the task is more difficult): $M. \rightarrow S.l. \rightarrow D.l.$. This way, the latter predicted labels should benefit from already having available the labels predicted before. This is one of the reasons we employed a sequence-to-sequence generative model for our classification tasks, our main focus being the **S.l.** and **D.l.** tasks, as for these the baseline model seems to struggle more. The difference between the architecture of the baseline model and the joint model is illustrated in Figure 1.

4.2 Additional Fine-tuning Task

Although not among the target tasks, we considered using the oven settings as an additional task in the joint model, only during fine-tuning, with the highest precedence compared to the target tasks: $OvenSettings \rightarrow M. \rightarrow S.l. \rightarrow D.l.$ The intuition was that the oven settings should influence the most the outcome (especially for **D.l.** and **S.l.** tasks, which should also be affected by whether the meat is the main content of the dish). The additional task of predicting the oven settings is formulated as predicting the triplet (T, t, P):

- Temperature setting (either in ^oC or ^oF)
- Time setting (in minutes)
- Cooking Program

Some of the recipes in the available dataset contain multiple such triplets, either because they are multi-step (in which case the triplets are joined with "AND"), or because the recipe provides alternative cooking instructions (pairs of triplets joined by an "OR"), we applied a pre-processing procedure, keeping at most five steps and only one of the options in each disjunction (see Table 1 for details):

$$\bigwedge_i (a_i \vee b_i) = \bigwedge_j c_j, j = min(i, 5)$$

where i is the number of steps.

This procedure was applied to reduce the number of tokens used for this task, to not use too many

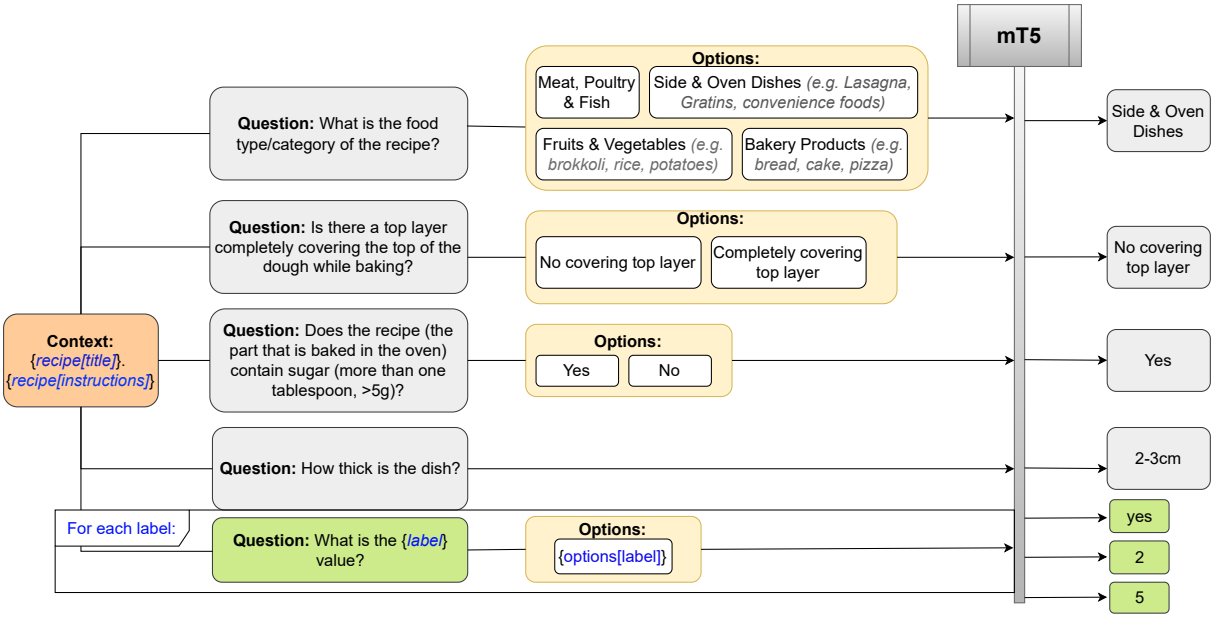

Figure 2: Question-driven Generative Framework with an example of question templates used only for training (gray boxes/first 4) and questions for the main tasks (**M.**, **S.l.** and **D.l.**) used for testing (green box/last one), with the corresponding options, and their explanatory information (gray text in parentheses).

tokens from the main tasks. We reduced the number of tokens used for the output of this task to 113 maximum. The percentage of entries in this dataset affected are (per language): De 9%, En 57%, Nl 1%, Fr 7%, It 15%, Sv 3%.

This model would be referred to as **OvenSettings**.

4.3 Question-driven Generative Framework (QdGF)

The central part of our approach is the Question-driven Generative Framework (**QdGF**), which uses a template to generate different types of questions from auxiliary/redundant training data. Inspired from Wu et al. (2022), our approach generates the response to the questions from auxiliary data available only during training and does not extract it from the target labels. To this extent, the model is fine-tuned in a multi-task manner, but with a variable number of tasks, as some of the additional information is not available for all the recipes in the training data. The only tasks to be guaranteed for all recipes are the main tasks, transformed in the same manner as the other tasks, to questions. The data is constructed in the following format: "<s>Context: $\{recipe[title]\}$. $\{recipe[instructions]\}$</s> Question: $\{Q_i\}$? Options: $\{options_i + explanations_i\}$</s>".

The number of questions/instructional prompts (i) used ranges between 6 and 10 per experiment. All these questions represent a subset from a set of 14 extracted questions initially. That is, we concatenate more information from the dataset (the recipe's title and instructions, together with several questions from the set of used questions, and their **answer options**, provided that the recipe is labeled with those annotations in the dataset), and use the resulting string as input to the mT5 model in the fine-tuning stage.

With these new prompts added, we try to stimulate the prior general knowledge of the model (Han et al., 2021) and direct it to a greater focus on this downstream task composed of the 3 target problems (**M.**, **S.l.**, and **D.l.**). Several elements in the options list also contain additional explanations that help guide the model, but these were removed from the labeled option. Providing the options to choose from and some related explanations (where available) should ground the model to the current problem domain and prevent potential hallucinations (Ahn et al., 2022).

A discussion on how the subset of used questions was chosen from the main set of questions is presented in Section 5.5 and an example of a possible instantiation of the framework is illustrated in Figure 2 (the best architecture used is similar to this, but with joint training for the three main tasks).

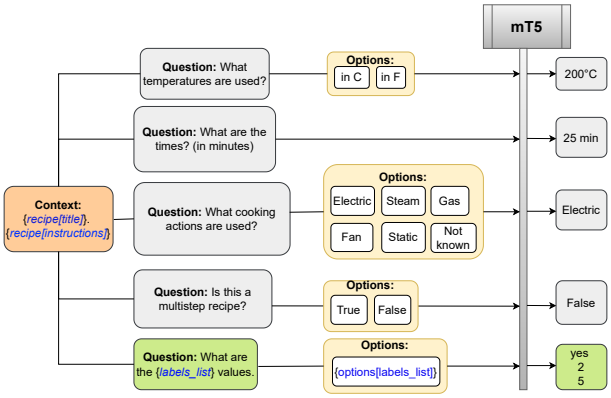

Figure 3: QdGF instantiated with the subset of oven settings questions.

4.3.1 QdGF + OvenSettings

We transformed the triplet (T, t, P), from Section 4.2, into 3 more questions, one for each oven setting, to further exploit the previous approach and adapt it to this framework. We also added an additional question, to make a difference between multi-step recipes (having more such triplets) and single-step recipes (with just one triplet). An example framework with this subset of questions is illustrated in Figure 3. The total number of potential additional questions to choose from, including the oven settings-related questions, is now 18.

All the extracted questions are available in Appendix B.

4.3.2 Renaming

One more potential issue we tried to overcome in this framework is related to the use of numbers to encode the class labels, which bears either no, or a potentially wrong semantic meaning for the model (Spokoyny et al., 2022). To address it, we changed each class from its numeric counterpart to a textual description consisting of 1-3 words, which we considered to best capture the class meaning. For example, for **S.l.**, the resulting mapping to textual classes is: $1 \rightarrow Maillard; 2 \rightarrow Caramelization$. The full mappings can be found in Appendix A. We integrated these newly renamed labels directly into QdGF, with their respective explanations in the **Options** component.

5 Experiments and Results

The dataset used contains approximately 52000 multilingual recipes labeled with the three target tasks and several additional annotations (few of them being sparse), performed by human experts

Lang.	M.	S.l.	D.l.
German	12:1	2:1	12:6:4:2:1
English	9:1	2:1	10:6:5:1:1
Dutch	7:1	1:1	10:8:8:3:1
French	12:1	2:1	4:2:2:1:1
Italian	17:1	2:1	7:4:4:4:1
Swedish	13:1	2:1	7:6:4:3:1
38:16:8:4:3:1	10:1	2:1	8:5:3:1:1

Table 2: Imbalance ratios per language (first column, last row), imbalance ratios per language and per task (with 2, 2, and 5 classes respectively) and imbalance ratios per task (last row, last three columns).

(see Section 3). It is highly imbalanced both respective to the languages and the classes used (see Table 2).

The train-validation-test split is $0.64 - 0.20 - 0.16$ for German and English (the fine-tuning languages). We fine-tuned only with the high-resource languages. The other languages are used in a 0-shot setting and also in a few-shot similar setting, due to the limited samples (Dutch 6%, French 2.6%, Italian 1.9%, Swedish 0.8%-shot setting respectively).

The backbone model employed is mT5 (Xue et al., 2020), a multilingual encoder-decoder transformer, and an attention-based model (Vaswani et al., 2017). Through manual hyperparameter tuning, we found the following hyperparameter values to produce the best results (and were used further): *Learning rate:* $3e-4$, *Weight decay:* 0.1, *No warmup* with Optimizer *AdaFactor* and Scheduler: *AdafactorSchedule*. Initially, we used AdamW Optimizer (Loshchilov and Hutter, 2017) and the linear scheduler with warmup from HuggingFace, but we changed to the original optimizer used for the T5 model (AdaFactor, Shazeer and Stern, 2018) which produced better results.

5.1 Specialization Pre-training

Initially, for all approaches, we used the mT5 model pre-trained by Google[1] on mC4. To further "specialize" the model on culinary data, we decided to further pre-train it on a multilingual dataset of approximately 490000 unlabeled recipes. As this dataset was not comparable in size to the original training corpus, pre-training the model from scratch using only these recipes did not provide enough information for the model to acquire a general semantic knowledge of words. Hence,

[1] https://huggingface.co/google/mt5-base

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Baseline	base	95.82	94.61	91.65	91.41	94.17	91.47	94.68
	large	97.26	96.83	92.57	95.91	95.63	95	96.43
Joint	large	97.3	**97.09**	93.78	95	**98.43**	93.38	96.7
Pt Joint	base	96.83	96.36	**95.51**	**96.59**	**97.81**	**96.69**	96.59
QdGF*	large	97.92	96.67	**94.46**	97.06	96.81	95.33	**96.98**
Pt QdGF*	base	**98.21**	**96.78**	93.97	96.56	95.84	96.33	**96.98**
	large	**97.96**	96.52	93.88	**97.17**	96.95	**97**	96.97

Table 3: **M.** - few-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Baseline	base	88.65	86.76	80.43	57.82	64.48	72.5	84.44
	large	90.88	89.62	71.25	70.68	70.4	53.33	85.87
Pt Joint	base	91.08	91.33	86.75	89.09	**90**	78.51	90.3
QdGF*	base	90.93	**92.94**	86.63	87.26	87.1	**88**	90.03
	large	90.15	91.22	**86.22**	**89.89**	**85.99**	85.66	89.39
Pt QdGF*	base	**91.86**	92.52	87.04	89.38	87.38	87.66	**90.66**
	large	**91.17**	**91.95**	85.82	87.56	85.57	**87.33**	**89.74**
Pt QdGF renam.*	base	91.28	90.75	**89.08**	**90.18**	87.81	87.33	90.4

Table 4: **S.l.** - few-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

the specialization pre-training was performed on the already pre-trained on mC4 model. We considered different approaches of pre-training, between Masked Language Modelling and Next Sentence Prediction, as Sun et al. (2022) highlights the reintroduction of NSP as a pre-training approach and also the importance of the pre-training corpus. However, we choose the original pre-training approach of the T5 model (Text-To-Text Transfer Transformer, Raffel et al., 2020), using a script performing span-masked language modeling[2].

5.2 Size Does(n't) Matter?

We tested all the sizes of mT5 that we were able to fit within our resources: *small*, *base*, and *large*, but focusing only on *base* and *large* as *small* did not stand up to the complexity of the tasks. For the first methods, *large* seemed to perform slightly better than *base* overall, but most importantly in the few-shot and 0-shot setting for the underrepresented languages. With the approach of incorporating the QdGF though, *base* performed similarly, or even slightly better than *large*, hinting that exploitation of hidden correlations using the right questions might be more important than the size and number of parameters of the model. Maybe this framework reduces the need for memorization, where large language models tend to be better (Tirumala et al., 2022). However, this claim is limited by the fact that, as the architecture grew more complex, the input data needed to be truncated for the *large* model to fit into available memory, thus maybe affecting performance. More on this in Limitations. We emphasize that our goal was not to compare between the two sizes, but to observe the improvements our method brings to each model (size) independently.

5.3 Overall Results

For assessing results, we considered the best models per problem out of the following (trained on De & En, few-shot setting for Nl, Fr, It, Sv):

- **Joint** (all three problems at once, fine-tuned on all languages, see Section 4.1).
- **OvenSettings** (joint model with the additional fine-tuning task, see Section 4.2).
- **QdGF** (model with a Question-driven Generative Framework, see Section 4.3), only the best instantiation of the questions is reported in the final results. Results from other instantiations are discussed in Section 5.5.

[2] https://github.com/huggingface/transformers/blob/main/examples/flax/language-modeling/run_t5_mlm_flax.py

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Baseline	base	79.54	73.46	66.8	60.35	55.38	59.69	74.28
	large	86.51	86.71	68.4	64.55	57.01	59.17	81.71
Pt Joint	base	87.17	86.5	82.26	74.54	72.18	72.72	**84.98**
Pt OvenSettings*	base	87.28	83.75	**82.77**	**78.91**	**74.45**	**86.44**	**84.98**
	large	86.98	84.58	**82.34**	**80.04**	**73.21**	**85.59**	**84.96**
Pt QdGF*	base	87.37	**87.28**	81.25	75.12	72.54	81.66	84.04
Pt QdGF+OvenSettings*	base	**87.77**	86.13	81.25	75.22	70.87	82	83.9
	large	**87.68**	**87.12**	80.93	77.65	68.51	78.66	83.98

Table 5: **D.l.** - few-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

Model	Size	De	En	Nl	Fr	It	Sv
Baseline	base	88 / 84	91 / **90**	22 / 84	0 / 78	0 / 55	0 / 60
	large	89 / 86	94 / **92**	34 / 85	0 / 81	0 / 53	0 / 50
Pt OvenSettings*	base	**91** / 81	**95** / 81	**90** / 85	**61** / **85**	**76** / 55	**97** / **83**
	large	90 / 82	93 / 83	**90** / 86	**63** / **87**	**75** / **61**	94 / **91**
Pt QdGF+OvenSettings*	base	**91** / **86**	84 / 89	**90** / **87**	55 / 83	70 / **67**	94 / 76
	large	**91** / **88**	**98** / 88	83 / **88**	49 / 85	66 / 59	**97** / 75

Table 6: **D.l.** - F1-scores for the 2 most underrepresented classes out of the 5 classes (2ndUnderrepClass / mostUnderrepClass). * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

- **QdGF+OvenSettings** (the model with a Question-driven Generative Framework that includes the question obtained from the additional task, see 4.3.1).
- **QdGF renam.** (the model with the renaming of classes for **S.l.** and **D.l.** tasks, see 4.3.2).

The overall results were considered per language, and for all languages combined using the weighted accuracy (with respect to the number of samples of each language). We also considered a mix of joint or not joint and specialized pre-trained or just default pre-trained. The baseline used to compare our results is one model per problem fine-tuned on all languages with a simple instructive prompt (dubbed "Baseline", see Section 3). We report the average result of multiple runs (with standard deviations up to 0.1%) for every mentioned model.

The final results for **M.** can be seen in Table 3. The best models based on the weighted accuracy are both QdGF Joint, one specialized pre-trained (with an increase of 2.3% compared to *base* Baseline) and one not. Based on the accuracy on the underrepresented languages, the specialized pre-trained Joint model performs the best, with an increase of 5.22% for Swedish (in a *base* model). We did not expect the renaming to help for this task since it was not applied to the classes of this problem (the classes already being "yes" or "no").

For **S.l.** (Table 4), the highest accuracy has been obtained by specialized QdGF joint again (increase of 6.22% in base). In this task, renaming (the second best) helped a lot, especially for the French language, with an increase of 32.36%. Also, the specialized joint model performed well (especially in Italian, with an increase of 25.52%), proving that this problem benefits from being joined with **M.**.

We expected oven settings to be related to **D.l.**, therefore the best models are the ones involving these auxiliary settings, as can be seen in Table 5. The specialized OvenSettings Joint model performed the best in both *base* and *large* and both, weighted accuracy and accuracy of every underrepresented language. This task also benefits from being joined with the other previous tasks. Specialized QdGF joint, which is the best model for the other two problems, is also one of the best here, followed by its extension with the settings auxiliary questions in weighted accuracy (with its German and English results being one of the best).

The QdGF approaches also helped to recognize the lowest represented classes, as the dataset is highly imbalanced, especially for **D.l.** task (Table 2, last column). For this, we measured the F1-score for the 2 most underrepresented classes in this problem, and the comparison can be seen in

Task	thickness	top	appearance	type	cheese	sugar	dough	dishes	temp
S.l.	-0.02	-0.03	0.17	-0.21	**1.00**	**1.00**	-0.49	-0.21	-0.10
D.l.	0.05	0.75	-0.84	-0.20	-0.01	0.25	**-1.00**	**1.00**	0.01

Table 7: Relevant correlations between **S.l.** and **D.l.** and answers of selected questions (Pearson correlation was used).

Table 6. A reason for the performance discrepancies in Table 6 - *Baseline row* between the second most underrepresented class (with an F1-score of 0 for underrepresented languages) and the most underrepresented class (with F1-score > 50) might be the model overfits on De & En (thus a higher F1-score for that class for De & En) and is unable to generalize to the other languages. We can see that employing Joint learning with OvenSettings and even QdGF, reduces the overfitting significantly.

Initially, we tested all the models (trained on De & En) in the 0-shot setting for the underrepresented languages (Nl, Fr, It, Sv). The model performed a bit better in these underrepresented languages after seeing examples from them. However, in some cases, it performed slightly worse in De & En, but nothing significant (in some cases it even performed better), implying that no catastrophic forgetting took place and that our method is able to generalize well even in new languages. These evaluations, together with the complete evaluations in the few-shot setting can be seen in Appendix C.

5.4 Domain generalizability

The recipe data used exhibit significant diversity in terms of structure and format, which means the model may not simply learn the specifics of the cooking recipes themselves. This, and the fact that specialization pre-training in the cooking domain did not help that much, suggests that the method can be generalized to other domains, by instantiating a QdGF with questions obtained from additional/redundant data, extracted data from the text to classify, or even metadata available in a new dataset/domain.

5.5 Ablation Study

The set of 18 available questions (see Appendix B) would result in 2^{18} possibilities of instantiations of the QdGF. To select the subset of questions, we checked for the most sparse questions among the dataset to try to avoid them, and we checked the correlations of the answers to the questions with our three main problems. We can see in Table 7 that the most correlated types of questions with the **S.l.** problem are *cheese* and *sugar*, which are contained in the best model for this problem (which is a **Pt QdGF*** with final acc. 90.66%). A decrease of 2.3% occurs in weighted accuracy few-shot, 5.22% in 0-shot, and 15.56% 0-shot for an underrepresented language, if discarding one of them. For **D.l.**, the most correlated questions (*dough*, *dishes*) are also part of the best QdGF model for this problem (Pt QdGF* with 84.04% and this one+OvenSettings* with 83.9%). The best versions of QdGF models in the final evaluations (Appendix C) differ between **S.l.** and **D.l.** by the subset of questions used, only the best were presented for each. All chosen subsets contain the *type* question as this correlates approx. -0.20 with **S.l.** and **D.l.** and 0.32 with **M.**.

6 Conclusion

Our study demonstrates the effectiveness of leveraging additional training data as instructional prompts in a multilingual, multitask classification problem, by introducing QdGF (Question-driven Generative Framework). Our proposed method achieves notable improvements in weighted multilingual accuracy, with absolute improvements of 2.3%, 6.22%, and 10.7% for the targeted classification tasks. Notably, the additional tasks related to oven settings and the highly correlated ones with the specific problems have the most significant impact. We observe that the size of the model and in-domain pre-training have minimal impact on final performance. Our findings underscore the importance of thoughtful training data selection and questioning strategies, particularly in underrepresented languages and imbalanced datasets. In such cases, we achieved substantial accuracy increases of 34.8% in the few-shot setting and 30.33% in the 0-shot setting, and a 97% increase in the F1-score of underrepresented classes, for the most underrepresented language. These results highlight the potential of leveraging additional training data and prompt-engineering to improve performance on multilingual, multitask models in text classifications.

Limitations

Truncation

Due to limited resources (a system with multiple GPUs NVIDIA Tesla V100 SXM2 16 GB, from which we used on average 3 GPUs per run), we were unable to run larger models than *mt5-large* (1,2B+ parameters). Regarding the maximum number of tokens accepted, for *base*, we used 512 tokens. Thus, recipes with more than 512 tokens were truncated. Approximately 500/55000 recipes were affected. As the architecture grew more complex, *large* became not as reliable as *base* due to the extra truncation needed. For the *large* model, we had to set the maximum number of tokens to 320, approx. 2600 recipes being affected, and for the more complex approaches (such as QdGF), we had to set the maximum number of tokens to 254, 5100+ recipes being affected, thus maybe affecting performance (as we can see in the latest tables, that results start to drop for *large* models as they get more complex). We did not test specifically how much truncation affects performance. Our main focus was the *base* model and we did not want to interfere with its truncation so we can measure this method's performance increase. A deeper study on what is the optimal truncation could be done in future. Setting the tokens to the same size to compare *base* vs *large*, to see which one generalizes better with our method, constitutes another future research interest, as this focused only on the improvements brought by the method to each model (size) individually. Therefore, access to more resources/better memory usage would make our Framework scalable to longer text also (longer than what the memory can fit, per model size), as it would not need this truncation.

Backbone model

We did not use other encoder-decoder models because we wanted to highlight the improvement this method brings to a language model, and we chose to show it on mT5, thus we selected mT5 (without the QdGF framework) as a baseline. Having other models as baselines constitutes another research interest of ours, depending on the available resources. The high computational cost and resources associated to larger and newer models were the main reasons we were unable to test our method on such other models. When it comes to LLM APIs, most of the services incur additional costs for fine-tuning, that were not available to our study. Moreover, this would raise concerns about the security and privacy of our data, since the dataset used is private.

Computational expense

Another resource-related limitation would be that our framework is computationally expensive, as, for every question, we replicate the recipe for that question (only if an answer is available). For a framework with 6 questions, the training data can grow up to 6 times, per epoch. This also increases the computation time proportionally. To solve this, we could take advantage of the context of LLMs, by prompting first the recipe, followed by the questions, without replicating the recipe each time. This would require a larger model. Another solution would be to feed the model a larger prompt composed of the recipe and all questions plus the desired tasks at once, but this again has the same downsize as mentioned before. Given the size of the output, this can also be susceptible to hallucinations, which can impact performance.

Data

Our method proved to work for: German, English, Dutch, French, Italian, and Swedish. This method might not work in languages not supported by the multilingual model. For new underrepresented languages (supported by mT5) our model will bring significant improvements in a 0-shot setting as our experiments showed. For the few-shot setting though, some additional annotations might need to be added/extracted manually for maximum improvement, which requires additional labor (although the effort might not be worth it, as results in the 0-shot setting are not much smaller).

We discovered a few wrongly labeled recipes in the dataset and a few recipes not properly scrapped from Web (containing HTML tags in instructions). We solved them, but we cannot guarantee that the final dataset was 100% clean and with no noise that might have affected performance, but we can say that QdGF adds some robustness to such noise, as the model "answers" some questions before making a final decision.

Our method is tested currently only on one unique dataset. We would like to emphasize that the dataset we used is relevant enough, both in terms of the amount of data, languages, and additional questions, especially for fine-tuning and 0-shot scenarios. However, the dataset is specific to the cooking domain. Applying a QdGF approach to data and problems from different domains might need data pre-processing and extraction of questions.

Ethics Statement

An important ethical concern is high energy consumption. As discussed in Limitations, our method requires computational resources that are avid energy consumers. A significant amount of electricity was consumed for running experiments, taking on average 3 hours per fine-tuning experiment (on 3 GPUs) and 12 hours per pre-training (on 4 GPUs).

Another ethical concern is the potential bias that may be present in the training data used to fine-tune the model, as the data were scrapped from public websites. It is essential to ensure that the dataset is free from misleading content.

Additionally, when deploying the model in real-world applications, it is crucial to consider the impact of classification errors or misinterpretations, particularly in kitchen safety domains, as various oven settings are involved.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. 2022. Do as i can, not as i say: Grounding language in robotic affordances.

Hadeel Al-Negheimish, Pranava Madhyastha, and Alessandra Russo. 2021. Numerical reasoning in machine reading comprehension tasks: are we there yet?

Shijie Chen, Yu Zhang, and Qiang Yang. 2021. Multi-task learning in natural language processing: An overview.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification.

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, Dennis Ulmer, Florian Schottmann, Khuyagbaatar Batsuren, Kaiser Sun, Koustuv Sinha, Leila Khalatbari, Maria Ryskina, Rita Frieske, Ryan Cotterell, and Zhijing Jin. 2023. State-of-the-art generalisation research in nlp: A taxonomy and review.

Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2021. A survey on text classification: From shallow to deep learning.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.

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.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Alex-Mihai Lăpuşan, Rareş-Liviu Horge, Sara Petres, Mihaela Dînşoreanu, Rodica Potolea, and Camelia Lemnaru. 2022. Instructions are all you need: Cooking parameters classification for monolingual recipes. In *2022 IEEE 18th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 73–80.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost.

Daniel Spokoyny, Chien-Sheng Wu, and Caiming Xiong. 2022. Numerical correlation in text. In *Proceedings of the 1st Workshop on Mathematical Natural Language Processing (MathNLP)*, pages 33–39, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Yi Sun, Yu Zheng, Chao Hao, and Hangping Qiu. 2022. Nsp-bert: A prompt-based few-shot learner through an original pre-training task–next sentence prediction.

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Yangjun Wu, Han Wang, Dongxiang Zhang, Gang Chen, and Hao Zhang. 2022. Incorporating instructional prompts into a unified generative framework for joint multiple intent detection and slot filling. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 7203–7208, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer.

Appendix

A Renamings

A.1 Surface aspect level (S.l.)

- $1 \rightarrow Maillard$
- $2 \rightarrow Caramelization.$

A.2 Dehydration level (D.l.)

- $1 \rightarrow Maintain\ elasticity$
- $2 \rightarrow Not\ too\ moist$
- $3 \rightarrow Rising\ dough$
- $4 \rightarrow Steam,\ dry,\ grill$
- $5 \rightarrow Crispy\ bottom.$

B The set of Questions

Each question is identified by its ID. Some answer options might contain additional explications (enclosed in parentheses).

B.1 Top

Is there a top layer completely covering the top of the dough while baking? *Options*:

- No covering top layer
- Completely covering top layer

B.2 Browned

Shall your food be browned on top? *Options*:

- Yes
- No

B.3 Appearance

How would you describe the appearance of the food? *Options*:

- One large
  (e.g. Lasagna, gratins, casseroles, etc.)
- Few thick items
  (grilled, stuffed foods)
- Many small items
  (e.g. french fries)
- One large thin item with crispy bottom
  (e.g. Quiche, Pizza, Tarte)

B.4 Prepare

How do you want to prepare the food? *Options*:

- Roasting one or few large pieces with a crispy surface
  (e.g. whole chicken or a roast)
- Cooking and baking of a casserole in a large container
- Cheese on top
- Airfrying many smaller pieces with a crispy surface
  (e.g. chicken legs, chicken wings)

B.5 Marinade

Do you use a sweet rub or marinade (e.g. honey rub or rub with brown sugar)? *Options*:

- Yes
- No

B.6 Cheese

Does the recipe use one of the following cheeses (Mozzarella, Pizza cheese, Gratin cheese)? *Options*:

- Yes
- No

B.7 Sugar

Does the recipe (the part that is baked in the oven) contain sugar (more than one tablespoon, >5g)? *Options*:

- Yes
- No

B.8 Dough

What type of dough/batter is it? *Options*:

- Any other
  (e.g with baking powder, not sure)
- Yeast and Bread doughs
- Puff Pastry

B.9 Dishes

Is it one of the three dishes: Pizza, Quiche(s) or Tarte(s)? *Options*:

- Yes
- No

B.10 Thick

How thick is the dish? Consider only the part placed in the oven. *Options*: in cm.

B.11 Pastry

Is the puff pastry filled with meat or fish? *Options*:

- Yes
- No

B.12 Type

What is the food type/category of the recipe? *Options*:

- Bakery Products
  (e.g. bread, cake, pizza)
- Side & Oven Dishes
  (e.g. Lasagna, Gratins, convenience foods)
- Fruits & Vegetables
  (e.g. broccoli, rice, potatoes, etc.)
- Meat, Poultry & Fish
- Not sure

B.13 Preheat

Does it need preheating? *Options*:

- Yes
- No

B.14 Thickness

How thick is the dish? *Options*:

- medium
- thin
- thick
- very thin

B.15 Multistep

Is this a multistep recipe? *Options*:

- True
- False

B.16 Temperature

What temperatures are used? *Options*: in C or F.

B.17 Time

What are the baking times? *Options*: in minutes.

B.18 Cooking_program

What is the cooking program used? *Options*:

- Electric
- Steam
- Gas
- Fan
- Static
- Heat
- Circulating air
- Grill
- Ventilated
- Hot air
- Convention
- Bottom
- Top heat
- Broil
- Top bottom
- Not sure

C Complete Evaluations

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Baseline	base	95.82	94.61	91.65	91.41	94.17	91.47	94.68
	large	97.26	96.83	92.57	95.91	95.63	95	96.43
Pt Joint	base	96.83	96.36	**95.51**	96.59	97.81	96.69	96.59
	large	96.17	96.62	92.86	95.68	96.25	94.21	95.83
Pt OvenSettings*	base	97.1	95.79	94.29	**97.96**	93.15	96.61	96.35
	large	96.7	95.53	92.85	96.82	96.26	95.72	95.95
QdGF	base	98.05	96.36	94.37	96.66	96.53	**97.66**	**96.98**
	large	96.66	95.79	94.37	**97.57**	96.67	95.66	96.26
QdGF*	base	97.67	96.15	93.8	97.06	**97.92**	96	96.79
	large	97.92	96.67	**94.46**	97.06	96.81	95.33	**96.98**
Pt QdGF*	base	**98.21**	**96.78**	93.97	96.56	95.84	96.33	**96.98**
	large	97.96	96.52	93.88	97.17	**96.95**	97	96.97
Pt QdGF+OvenSettings*	base	97.9	96.57	93.31	96.36	97.36	96	96.8
	large	**98.05**	96.1	93.8	97.27	96.81	**97.66**	96.94
Pt QdGF renam.	base	97.79	96.46	95.27	96.35	95.56	95	96.81
	large	97.43	**97.04**	**94.46**	96.76	96.53	94.66	96.75
Pt QdGF renam.*	base	97.23	96.62	95.27	96.15	95.56	95.33	96.57
	large	96.46	95.79	93.64	94.43	96.12	94.33	95.67

Table 8: **M.** - few-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Joint	base	95.95	95.69	90.01	92.72	94.37	95.86	94.95
	large	97.3	**97.09**	93.78	95	**98.43**	93.38	**96.7**
Pt Joint	base	97.3	**96.99**	92.86	93.86	94.37	96.69	**96.4**
	large	96.66	96.57	91.64	92.5	94.68	96.69	95.75
OvenSettings*	base	97.25	96.83	92.35	95.01	95.32	**98.3**	**96.4**
	large	97.17	95.29	92.75	**97.5**	94.7	94.91	96.1
Pt OvenSettings*	base	96.79	96.05	**93.57**	**97.73**	**95.95**	98.29	96.28
	large	97.03	96.78	**93.88**	96.6	96.88	**97.45**	96.58
QdGF	base	98.01	96.15	90.25	92.86	94.81	95.32	94.87
	large	97.7	96.83	92.66	95.95	96.53	96.33	96.58
QdGF*	base	97.72	96.31	91.36	94.44	95.56	96	96.08
	large	**98.12**	96.57	92.01	95.35	96.11	97	96.57
Pt QdGF*	base	**98.07**	96.88	91.44	94.03	94.03	96.33	96.23
	large	97.9	96.67	90.87	95.14	93.06	95.66	96.05
Pt QdGF+OvenSettings*	base	97.94	96.78	90.62	93.93	95	96.66	96.11
	large	97.79	96.31	90.62	94.23	95.28	95.33	95.96
Pt QdGF renam.	base	97.85	96.57	92.34	94.13	94.73	94.33	96.18
	large	97.43	96.72	91.11	94.33	93.9	94.66	95.83
Pt QdGF renam.*	base	97.61	96.78	91.36	95.64	94.6	94.66	96.13
	large	97.23	97.04	93.48	96.35	96.4	95.66	96.52

Table 9: **M.** - 0-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Baseline	base	88.65	86.76	80.43	57.82	64.48	72.5	84.44
	large	90.88	89.62	71.25	70.68	70.4	53.33	85.87
Pt Joint	base	91.08	91.33	86.75	89.09	**90**	78.51	90.3
	large	90.26	90.13	84.5	87.73	**88.12**	80.16	89.19
Pt OvenSettings*	base	89.78	89.15	86.54	**91.61**	86.29	88.13	89.19
	large	88.64	88.79	81.43	88.21	81.62	**88.13**	87.52
QdGF	base	90.88	91.85	85.73	87.46	86.4	**88.33**	89.66
	large	89.07	90.29	86.14	87.26	86.13	86.66	88.46
QdGF*	base	90.93	**92.94**	86.63	87.26	87.1	88	90.03
	large	90.68	91.69	**87.2**	87.36	86.37	86.33	89.65
Pt QdGF*	base	**91.86**	92.52	87.04	89.38	87.38	87.66	**90.66**
	large	**91.17**	**91.95**	85.82	87.56	85.57	87.33	**89.74**
Pt QdGF+OvenSettings*	base	91.66	92	87.04	90.49	86.68	83.33	90.39
	large	90.97	90.7	85.49	86.55	83.77	87.66	89.13
Pt QdGF renam.	base	89.14	90.96	88.02	88.96	87.39	88	89.18
	large	88.56	88.36	83.37	**88.56**	84.21	84.33	87.41
Pt QdGF renam.*	base	91.28	90.75	**89.08**	90.18	87.81	87.33	90.4
	large	86.18	85.82	81.42	82.89	81.16	83	84.69

Table 10: **S.l.** - few-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Joint	base	88.45	87.22	**81.54**	75.22	**82.81**	67.76	86.13
	large	**91.48**	92.21	**84.7**	80	**86.56**	68.59	**89.72**
Pt Joint	base	**91.95**	91.79	78.49	**82.95**	81.25	57.85	**88.94**
	large	89.95	90.81	77.67	46.36	66.25	59.5	85.04
OvenSettings*	base	90.73	88.58	74.92	62.81	77.88	73.73	86.14
	large	90.18	90.38	75.1	80.5	84.11	74.57	87.47
Pt OvenSettings*	base	90.95	**92.31**	74.61	75.28	76.94	70.94	87.68
	large	89.99	90.81	76.86	80.27	83.8	77.12	87.69
QdGF	base	91.48	92.26	77.31	65.47	61.27	70.89	81.72
	large	90.97	91.95	82.72	80.28	83.91	**83.66**	88.28
QdGF*	base	90.93	92.05	81.5	75.12	76.83	**79.66**	86.94
	large	90.8	91.9	81.74	**83.21**	83.08	79	88.15
Pt QdGF*	base	91.17	92	80.19	79.67	79.33	77.66	87.47
	large	90.91	**92.26**	75.14	80.79	71.15	70	86.02
Pt QdGF+OvenSettings*	base	91.13	91.74	75.71	78.76	78.5	77	86.66
	large	90.71	91.79	75.22	74.62	68.51	72	85.09
Pt QdGF renam.	base	90.36	91.79	55.74	68.11	48.61	43.33	79.42
	large	89.96	91.22	74.9	69.33	54.84	68.66	82.92
Pt QdGF renam.*	base	91.13	91.48	77.83	75.3	74.1	64.33	85.8
	large	87.02	87.58	78.16	78.44	77.14	81.66	84.23

Table 11: **S.l.** - 0-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Baseline	base	79.54	73.46	66.8	60.35	55.38	59.69	74.28
	large	86.51	86.71	68.4	64.55	57.01	59.17	81.71
Pt Joint	base	87.17	86.5	82.26	74.54	72.18	72.72	**84.98**
	large	86.19	85.51	79.61	72.5	70.31	71.9	83.71
Pt OvenSettings*	base	87.28	83.75	**82.77**	**78.91**	**74.45**	**86.44**	**84.98**
	large	86.98	84.58	**82.34**	**80.04**	**73.21**	**85.59**	**84.96**
QdGF	base	86.97	87.12	78.4	69.06	72.12	78	82.7
	large	87.32	86.55	80.27	72.09	70.59	80	83.25
QdGF*	base	87.26	86.76	80.6	72.29	70.18	80	83.29
	large	86.68	86.45	81.17	76.34	69.9	80.66	83.44
Pt QdGF*	base	87.37	**87.28**	81.25	75.12	72.54	81.66	84.04
	large	86.88	86.45	81.42	74.72	67.13	82.66	83.26
Pt QdGF+OvenSettings*	base	**87.77**	86.13	81.25	75.22	70.87	82	83.9
	large	**87.68**	**87.12**	80.93	77.65	68.51	78.66	83.98
Pt QdGF renam.	base	85.76	85.45	80.76	75.6	69.66	83.66	82.76
	large	85.4	84.46	78.48	73.78	69.25	79.66	81.77
Pt QdGF renam.*	base	86.71	84.83	82.72	70.44	65.65	79.66	82.38
	large	84.54	82.02	80.27	69.23	63.02	80.33	80.2

Table 12: **D.l.** - few-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.

Model	Size	De	En	Nl	Fr	It	Sv	Weighted Acc
Joint	base	83.5	81.62	63.3	57.72	52.81	48.76	77.62
	large	87.03	86.65	74	67.72	60	57.85	82.92
Pt Joint	base	**87.59**	85.36	73.09	69.32	56.56	52.89	82.69
	large	85.58	83.49	65.85	51.13	47.5	48.76	78.93
OvenSettings*	base	86.76	85.31	68.7	58.27	53.89	62.71	81.17
	large	86.75	84.51	74.79	70.52	59.19	70.34	82.66
Pt OvenSettings*	base	87.36	85.56	**76.96**	**70.52**	**60.12**	**70.08**	**83.52**
	large	87.2	84.41	76.76	**72.79**	60.43	73.73	**83.34**
QdGF	base	86.95	86.13	64.31	63.36	51.29	60.61	74.38
	large	**87.52**	85.25	70.82	66.53	53.4	68.66	79.69
QdGF*	base	87.37	**87.48**	70.25	64.71	51.04	62.66	79.44
	large	87.1	**87.9**	76.2	70.47	**63.1**	72.33	81.94
Pt QdGF*	base	87.3	87.22	69.84	66.63	57.83	65.33	80.09
	large	87.04	86.81	66.34	65.01	45.49	59	78.16
Pt QdGF+OvenSettings*	base	87.48	86.19	72.45	67.94	53.12	62.33	79.98
	large	87.23	85.46	65.93	65.82	50.48	62	78.48
Pt QdGF renam.	base	83.17	83.84	63.81	54.05	40.16	60.33	73.96
	large	84.1	84.2	68.86	61.64	52.21	66.66	76.98
Pt QdGF renam.*	base	87.02	86.28	72.21	69.33	55.54	67.33	80.23
	large	85.84	84.15	**77.18**	71.45	62.74	**75.33**	80.89

Table 13: **D.l.** - 0-shot setting accuracy for the underrepresented languages, trained on De & En. * = also joint models. Pt = specialized pre-trained. blue=Best *base*, teal=Best *large*.