

# MaChAmp at SemEval-2023 tasks 2, 3, 4, 5, 7, 8, 9, 10, 11, and 12: On the Effectiveness of Intermediate Training on an Uncurated Collection of Datasets.

**Rob van der Goot**

IT University of Copenhagen  
robv@itu.dk

## Abstract

To improve the ability of language models to handle Natural Language Processing (NLP) tasks and intermediate step of pre-training has recently been introduced. In this setup, one takes a pre-trained language model, trains it on a (set of) NLP dataset(s), and then finetunes it for a target task. It is known that the selection of relevant transfer tasks is important, but recently some work has shown substantial performance gains by doing intermediate training on a very large set of datasets. Most previous work uses generative language models or only focuses on one or a couple of tasks and uses a carefully curated setup. We compare intermediate training with one or many tasks in a setup where the choice of datasets is more arbitrary; we use all SemEval 2023 text-based tasks. We reach performance improvements for most tasks when using intermediate training. Gains are higher when doing intermediate training on single tasks than all tasks if the right transfer task is identified. Dataset smoothing and heterogeneous batching did not lead to robust gains in our setup.<sup>1</sup>

## 1 Introduction

The introduction of word embeddings and later contextualized transformer-based models, i.e. Large Language Models (LLM), have led to performance improvements on many Natural Language Processing tasks. Technically these approaches are multi-task learning approaches (with parameter sharing over time), where we first train a language model on raw data, save the weights, and then re-train the weights for our target task. It has been shown that intermediate steps of training can be beneficial for downstream performance. This intermediate step of training can be done through language modeling to adapt to new domains or languages (Gururangan et al., 2020; Muller et al.,

2021) or on NLP datasets directly (Phang et al., 2018; Aribandi et al., 2022), exploiting the synergies across tasks. Intermediate training on NLP datasets has benefits both for performance and efficiency (retraining on tasks is cheaper compared to training a full language model).

One important desideratum in finding the right language model for a downstream task is the training data and its distance to the target data. For an intermediate training step on NLP datasets this is even more complex, as there are more design decisions to make; e.g.: which NLP tasks are relevant for the target task? Which datasets are closest to the target data? How many (or how much) of them do we use? Which LLM to use as a starting point? Previous work has investigated mostly single dimensions in this choice of intermediate training datasets (in a carefully curated setup), less is known about what to do if one has a more varied set of NLP tasks.

In this paper, we use all text-based tasks of SemEval 2023 (tasks 2-12) as a seemingly arbitrary set of NLP tasks for evaluating the effect of intermediate training. More concretely, we seek to answer:

- Is the selection of the target or the source task more important for successful transfer?
- How does training on a combination of tasks compare to intermediate training on single tasks?
- Are dataset smoothing or heterogeneous batches beneficial for intermediate training?
- Which properties of datasets and/or tasks are good predictors for performance gains for transfer learning?

We test all of these under a setting of highly varied datasets, in different languages, with different tasks, different task types (single labels, sequences, etc.), different training sizes, and predictions over different input lengths (word, sentence, document).

<sup>1</sup>Code available at: <https://bitbucket.org/robvanderg/semEval2023/>

## 2 Intermediate Training

An intermediate step of language model training has shown to be beneficial to adapt a language model to a new domain, like social media or biomedical data (Gururangan et al., 2020; Barbieri et al., 2022) as well as to new languages (Muller et al., 2021; Chau and Smith, 2021). Here, the intuition is to repurpose the knowledge of previously trained models, and specialize them towards the target domain and/or language by doing further finetuning on a language modeling objective.

Phang et al. (2018) showed that an intermediate training step on an NLP tasks can also be beneficial for performance of LLM’s on the GLUE datasets, which are all on the sentence level. Follow-up work attempted to identify how datasets can be selected for the intermediate training step. Wang et al. (2019) find that language modeling is hard to beat as intermediate task, and that multitask pre-training outperforms single-task pre-training. They also included sequence-to-sequence tasks, and conclude that these are too distant to be beneficial for classification tasks. Correlations on performance of dataset pairs are low showing that it is hard to predict which datasets are beneficial. This is confirmed by the findings of Chang and Lu (2021), who conclude that task complexity is not a good predictor for being a good transfer dataset, whereas Pruksachatkun et al. (2020) find the opposite (complex tasks are good to transfer from), although they do conclude that future work is necessary.

Weller et al. (2022) compare intermediate training versus joint training on sentence-level tasks, concluding that for small datasets, joint training is more beneficial, and for larger datasets intermediate training should be used. Poth et al. (2021) evaluate a variety of (supervised) approaches to automatically identify which source datasets are beneficial, and conclude that pre-computable sentence representations are efficient for this task, and confirm that within task-type transfer outperforms cross-task-type transfer which is in line with the findings of Padmakumar et al. (2022) who compare transfer across and within task types.

Instead of selecting tasks, recent work has attempted to train on a wide variety of tasks. This is commonly done in the space of generative language models, where training on a variety of tasks is easier because many NLP tasks can be converted to generation tasks, and can then directly be used to (re-)train an autoregressive language

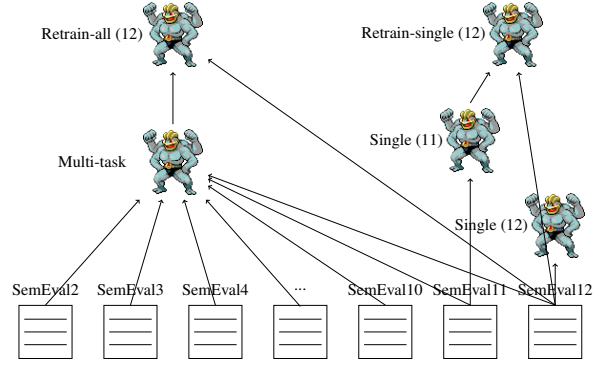



Figure 1: Overview of the setup with all models evaluated on task 12. Models are depicted by , and the target task is shown in brackets. Note that the multi-task model can output all SemEval tasks, and single (11) is included to better visualize retrain-single (12).

model (Aribandi et al., 2022; Sanh et al., 2022; Chung et al., 2022). In this setup it is easier to exploit a large variety of task types and a much higher amount of datasets (~50-1,800 datasets) is used compared to previous work. However, autoregressive language models still lag behind in performance compared to autoencoder language models for many tasks. Work that focused on intermediate training for encoder models and a large variety of task types is mainly done in the biomedical domain (Parmar et al., 2022; Fries et al., 2022).

Compared to previous work on autoencoder models, we have a larger variety of tasks as well as languages. Our selection of datasets is somewhat arbitrary (all SemEval 2023 tasks), leading to a more challenging setup for multi-task learning compared to previous work who usually used a carefully curated set of datasets (with often 1 language and/or task type). The most similar to our setup is van der Goot (2022), as they also use an arbitrary set of tasks (all SemEval 2022 text-based tasks) and also compare joint training with intermediate training on the full collection of the data. However, their results show no clear trend on when one approach outperforms the other. We build on this work by considering also single-task intermediate training, and systematically analyzing the differences in performance based on properties of the datasets. Furthermore, we evaluate the effect of diverse batching and dataset smoothing.

## 3 Setup

We first find the best strategy within each dataset (Section 4); we refer to these models as

2.	[johann adam birkenstock] <sub>Artist</sub> in 1774 founded [birkenstock] <sub>PrivateCorp</sub> shoe company
3-1.	Watch: Campus Commie Has Profanity-Laden Hissy Fit, Pours Beverage on FSU Republicans ... $\mapsto$ opinion
3-2.	Illegal alien wanted for attempted murder in NC arrested at US-Mexico border ... $\mapsto$ Crime_and_punishment Security_and_defense
3-3.	Online Sociology Course Founders Over Whether Australia is a Country ... $\mapsto$ Conversation_Killer Doubt
4.	We should ban whaling   against   whaling is quite a profitable profession. $\mapsto$ Achievement Power: resources
5-1.	This dude reckons you can lose weight on a diet of pies and beer $\mapsto$ passage
5-2.	A simple way to fight clickbait:   ... Don't like clickbait? [Don't click] <sub>spoiler</sub>
6-1.	[DATE: MARCH 15,] <sub>preamble</sub> [ORAL] <sub>preamble</sub> [(Per Akil Kureshi, J.)] <sub>none</sub> [The petitioner has challenged an order dated ...] <sub>fac</sub>
6-2.	[Section 46,] <sub>provision</sub> [Provincial Insolvency Act,] <sub>statute</sub> and [section 47,] <sub>provision</sub> [Presidency Towns Insolvency ACT,] <sub>statute</sub> deal with ...
6-3.	... words and who has retired on or after the 1st day of October, 1974 are unconstitutional and are struck down. ... $\mapsto$ denied
7-1.	Patients in NCT02953860 receive less mg of Enzalutamide than Fulvestrant on a weekly basis. $\mapsto$ Contradiction
7-2.	More than 1/3 of patients in cohort 1 of the primary trial experienced an adverse event.   Adverse Events 1: [Total: 69/258 (26.74%)] <sub>evidence</sub> ...
8-1.	[Colchicine toxicity?] <sub>question</sub> [Death?] <sub>question</sub> [Very scared] <sub>per_exp</sub> ...
8-2.	... [SLE] <sub>population</sub> , and/or [MCTD] <sub>population</sub> I have had horrible [body aches,] <sub>outcome</sub> [swelling] <sub>outcome</sub> and [fatigue] <sub>outcome</sub> almost daily
9.	@user Ohhhh, Google is struggling to translate it $\mapsto$ 1.4
10-1.	U sure ? Id personally never trust a girl from pluto $\mapsto$ not sexist
10-2.	The absolute state of women gentlemen $\mapsto$ 2. derogation
10-3.	Thank you for all the women who are still sensible. $\mapsto$ 3.3 backhanded gendered compliments
11-1.	BRITISH REFUGEES WELCOME #Brexit $\mapsto$ 0.0
11-2.	رخيصات مين باقي يقول $\mapsto$ 0.67
11-3.	prev_agent": "You are sure?", "prev_user": "omg yes", "agent": "You are sure?", "user": "foolish" $\mapsto$ 0.25
11-4.	life sucks when you're dishonest $\mapsto$ 0.4
11-4.	Why is Nevada only 67% of votes counted .... bloody slow pokes! #Elections2020 $\mapsto$ 0.2
12.	e dey always taste like paper and e no dey get oil $\mapsto$ negative

Table 1: Example input and annotation for each task. The ellipsis indicate that the utterance continues.  $\mapsto$  separates input and output. The vertical bar (|) is used to separate multiple inputs or outputs. Spans are represented with square brackets and the label in subscript.

Name	Subtasks	Languages	Size
2. MultiCoNER II	NER	BN, DE, EN, ES, FA, FR, HI, IT, PT, SV, UK, ZH	2,672,490
3. News persuasion	1. News categorization 2. Framing classification 3. Persuasion technique classification	EN, FR, GE, IT, PO, RU EN, FR, GE, IT, PO, RU EN, FR, GE, IT, PO, RU	741,561 725,740 19,561,550
4. ValueEval	Human value classification	EN	116,294
5. Clickbait spoiling	1. Spoiler type classification 2. Spoiler detection	EN EN	34,520 1,647,176
6. LegalEval	1. Rhetorical role detection 2. NER 3. Legal judgement prediction	EN EN EN	755,280 369,205 5,082
7. Clinical NLI	1. Entailment 2. Evidence retrieval	EN EN	21,828 311,687
8. Medical claims	1. Claim identification 2. PIO frame extraction	EN EN	549,231 78,864
9. Tweet intimacy	Intimacy Analysis	EN, ES, IT, PO, FR, ZH	73,698
10. Explainable sexism	1. Sexism detection 2. Sexism classification 3. Fine-grained sexism classification	EN EN EN	262,939 68,043 68,043
11. Le-Wi-Di	1. Hate speech detection* 2. Misogyny detection* 3. Abuse detection* 4. Offensiveness detection*	AR EN EN EN	14,252 12,788 64,738 145,245
12. AfriSenti-SemEval	Sentiment classification	AM, DZ, HA, IG, KR, MA, PCM, PT, SW, TS, TWI, YO	795,449

Table 2: Overview of tasks and their data. Data size is here represented as number of words in (the labeled part of the) training data as counted with `wc` (whitespace-based). Names of tasks are shortened to fit in the table, subsection titles of Section 4 include the full names of the tasks. \*for task 11, soft labels need to be predicted (which are the average over 5 annotators).

single. Then we use these best settings of each task to train a single multi-task model covering all tasks (multi-task). We re-train this multi-task model on each target task separately (retrain-all), which is the intermediate training setup described in Section 2. We compare this to a setup where we re-train from single task models (retrain-single), where we focus mostly on the best source transfer task. A schematic over-

view of the setup is shown in Figure 1.

For the multi-task models, we evaluate the effect of dataset smoothing, where we use multinomial smoothing with a factor of 0.5. We also experiment with task-diverse batches (i.e. heterogeneous batching: multiple tasks/datasets in a single batch), which has previously shown to be beneficial for intermediate training of generative models (Aghajanyan et al., 2021).

We use MaChAmp (van der Goot, 2022) v0.4 with default hyperparameters for all experiments, with the bert-base-multilingual-cased language model for efficiency reasons. For our final test submissions, we compare 7 language models on the best strategy for each task. We used a pre-selection of multilingual language models based on previous empirical results achieved by MaChAmp.<sup>2</sup> The list of language models and their results can be found in Appendix C.

## 4 Data and Baselines

For each task, we describe an overview of the task, the baseline approaches we evaluated, and the results. An annotated example for each task can be found in Table 1, and an overview with dataset statistics in Table 2. We followed the officially recommended metrics for each (sub)task; we used

<sup>2</sup>[https://robvanderg.github.io/blog/tune\\_lms.htm](https://robvanderg.github.io/blog/tune_lms.htm)

Lang	SL-bio	SL-seq	ML-bio-shared	ML-bio-sep	ML-seq-shared	ML-seq-sep
bn	77.07	75.95	81.03	79.50	76.99	79.07
de	69.03	67.09	75.87	74.17	74.28	73.05
en	67.13	65.18	71.80	71.40	69.95	70.54
es	71.08	69.70	77.08	76.05	75.06	75.54
fa	65.57	62.09	66.88	64.82	63.57	63.58
fr	71.58	69.60	76.13	74.90	74.56	74.30
hi	78.34	76.42	78.11	79.97	77.35	78.10
it	77.81	75.25	81.95	81.23	80.89	80.18
pt	72.37	68.21	76.44	76.57	72.78	72.44
sv	73.13	72.39	77.96	79.29	76.44	77.89
uk	70.96	70.04	72.54	71.37	72.74	72.42
zh	73.15	70.51	76.61	76.84	69.33	71.37
Avg.	72.27	70.20	76.03	75.51	73.66	74.04

Table 3: Results task2 (Span-F1 from conl1eval.pl)

the internal implementation of these metrics in MaChAmp for model selection when available. We use the Scikit-Learn implementation for f1 scores and `conl1eval.pl` for spans for the scores reported in this paper. For tasks without publicly available dev data (3,7,8,9,10, and 12), we use 80% for train and 20% for dev. Task 6 is described in Appendix B, as we did not manage to officially participate.

#### 4.1 Task 2: Multilingual Complex Named Entity Recognition (MultiCoNER 2)

Task 2 concerns multilingual named entity recognition. The data is characterized by its large size and the fact that the entity labels are fine-grained; in total 35 labels are used. The data is taken from Wikipedia, questions from the MS-MARCO QnA corpus (Nguyen et al., 2016), and search queries from ORCAS (Craswell et al., 2020), and is labeled using weak supervision (Malmasi et al., 2022).

In the basic setup, we use a simple feedforward layer on top of the encoder for word-level classification of the BIO labels (`seq`), we also experimented with a CRF layer that enforces valid BIO-sequences (`bio`). We further experimented with single language models (`SL`) and multilingual models (`ML`). In the multilingual setup, we distinguish between a model that shares the decoder across all languages (`shared`), and a model that trains a separate decoder for each language (`sep`).

Results (Table 3) show that the CRF layer is beneficial in all settings. Furthermore, sharing as many parameters as possible is beneficial; the multilingual model with separate decoder outperforms the mono-lingual baselines, but the highest scores are obtained when also sharing the decoder.

#### 4.2 Task3: Detecting the genre, the framing, and the persuasion techniques in online news in a multilingual setup

Task 3 includes three classification tasks on news articles (Piskorski et al., 2023). The inputs are relatively long (734-1210 words per article), which poses problems for current language models. The first subtask is genre classification, in which each article is classified as opinion, reporting, or satire. The second subtask is framing classification, which is a multi-label classification problem with 14 labels. The third subtask is framing technique classification, which is also multi-label, and has 23 labels. For the third subtask, there are also instances without any label, whereas for the second task, each instance has at least one label.

For the first subtask, we use a single feedforward layer to obtain predictions and use a cross-entropy loss. We experiment with single- (`SL`) and multilingual (`ML`) models, and for the multilingual models evaluate a shared decoder (`shared`), and separate language decoders (`sep`). For subtasks 2 and 3, we evaluate the same setups, and attempt to model the task in three different ways: first, we consider the multi-labels as if they are one label by concatenating them (`clas`). Secondly, we attempt to model them as separate tasks (`sep_clas`). Third, we use a multi-label setup (`multi_clas`) in which we use a BCE loss with a Sigmoid layer and manually set the threshold above which probability we output a label. For the classification tasks, only the first 128 subwords are used due to memory restrictions.

Table 13 (Appendix A) shows that the multilingual classification models perform well for subtasks 1 and 2. Results on subtask 3 are less stable, and multi-clas does better here after finding the optimal threshold. Sharing the decoder is beneficial for subtasks 1 and 3.

#### 4.3 Task 4: ValueEval: Identification of Human Values behind Arguments

Task 4 (Kiesel et al., 2023) is a classification task for arguments; they are to be classified in one or multiple of 20 human values, which are described in Kiesel et al. (2022). The input consists of a conclusion, the premise’s stance towards the conclusion, and the premise itself. We include all three texts with a special `SEP` token as divider and give the subwords in the stance segment ID’s (Devlin et al., 2019) of 1.

Because this task is a multi-label classification problem, we compare three approaches: 1) con-



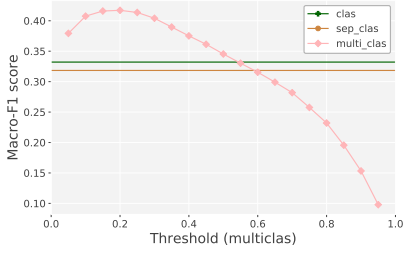


Figure 2: Results of task 4 (F1): separate classifiers (sep), a joint classifier (all), and a multi-clas classifier with a threshold (multiclas).

sider the combination of classes as one label (`clas`) 2) train 20 separate (binary) decoders, one for each label (`sep_clas`) 3) train a multi-head classifier, which uses a BCE loss with a Sigmoid layer and then outputs all labels above a certain threshold (`multi_clas`). We also evaluated whether concatenating the data from the Zhihu dataset (Kiesel et al., 2022) to the training data was beneficial, but saw lower scores across all approaches.

Results in Figure 2 show that training separate classifiers for each label is not beneficial, simply concatenating labels leads to higher performance. However, training a multi-head classifier and tuning the threshold leads to superior performance, where tuning the threshold is crucial.

#### 4.4 Task 5: Clickbait Spoiling

Task 5 concerns the classification (subtask 1) and spoiler detection (subtask 2) for clickbait posts and their corresponding full text. A clickbait post is a short text that is intended to inappropriately entice readers to visit a web page. The data was taken from Facebook, Reddit and Twitter (Fröbe et al., 2023; Hagen et al., 2022). For subtask 1, three types of spoilers are classified; phrase, multi, and passage. These labels refer to the type of extraction that is needed to find the spoilers, phrase means that the spoiler is just a short phrase, passage refers to spoilers consisting of subsequent phrases, and multi to more complex spoilers (combination of phrases throughout the document). Subtask two concerns the extraction of spoilers in the text from the web page, this is done by locating a (series of) span(s).

For subtask 1, we identify which pieces of information from the dataset are useful for the target labels. We include the text of the post, the description, title text, keywords, links of media, and URL of the webpage. We tried each of these inputs in isolation and found that all of them beat the major-

5-1 Information	F1			
Post text	66.92	5-2	Span-F1	
Page description	47.73			
Page title	59.12	seq	15.37	
Page text	47.85			
Page keywords	43.26	bio	17.21	
media	38.04			
url	51.08			
all	67.24			
all-media	66.40			

Table 4: Results for task 5, subtask 1 (Macro F1) and subtask 2 (span-f1)

ity baseline, so our final model includes all inputs. For subtask 2, we convert the character offsets of the spans to BIO labels on the word level and evaluate a sequence labeler with and without CRF layer. Note that this does not take the clickbait post into account, and is thus a non-realistic setup.

Results show that the original clickbait text (Post text) is the most predictive source of information in isolation. However, each of the used information categories outperform the majority baseline (19.68). Because of these positive results, we also train a model with all categories, which slightly outperforms using only the post text. After ablating the least informative category (media) from the “all” model, we observe a performance drop and decided to stick with the full combined model. For subtask 2, the CRF layer with BIO constraints is superior once again.

#### 4.5 Task 7: Multi-Evidence Natural Language Inference for Clinical Trial Data

Task 7 consists of data from clinical trial reports for breast cancer studies (Jullien et al., 2023). The first subtask is to distinguish entailment from contradiction based on a clinical trial report and a statement. The second subtask is to extract the spans of the supporting facts to justify the output of subtask 1. The facts are sentences, and the sentence boundaries are given.

We implement the first task as classification task. We tried to predict based on the statement, and additionally experiment with adding the section and/or the text. The second subtask we also implemented as a classification task, we include pairs of the statement and 1 sentence from the trial report at a time, with a binary label (relevant or not).

For subtask 1 The majority baseline would score 33.33 F1, so by just using the statement we can

7-1		7-2	
Classification	59.98	classification	68.21
+section	62.22		
+text	56.58		
+comb	52.94		

Table 5: Macro-f1 scores for subtask of task 7

subtask	seq	bio
8-1	23.58	32.91
8-2	28.47	28.48

Table 6: Span-F1 scores for task 8.

already obtain a substantially higher performance (Table 5). For subtask 2, the majority baseline would score 52.21, so the classification per sentence pair is effective.

#### 4.6 Task 8: Causal medical claim identification and related PICO frame extraction from social media posts

Task 8 (Khetan et al., 2023) is based on medical data from Reddit (Wadhwa et al., 2023). The first subtask is to identify spans that represent claims, experiences, experience-based claims, or questions. The second subtask is to find spans that identify the population, intervention, or outcome frame of a claim. It should be noted that the spans of the first task are much longer (average of 23 words) compared to the spans of the second subtask (average of 2 words). The text for task 8 was not released, but a scraping script was provided. We scraped the data on 07-11-2023, and after aligning the annotations we noticed that some posts were deleted or edited. For subtask 1, we removed 767 of the 5695 posts, and for subtask 2, 72 out of 597.

The results (Table 6) show a clear trend; for the longer spans of subtask 1, the sequence label approach with a feedforward layer is ineffective,

Language	SL	ML-shared	ML-sep
Chinese	62.88	65.22	64.12
English	71.37	71.26	69.96
French	54.63	53.47	56.18
Italian	54.13	59.37	59.25
Portuguese	60.07	61.46	60.28
Spanish	64.48	66.97	64.10
Avg.	61.26	62.96	62.32

Table 7: Pearson  $r$  scores for task 9 per language and average over languages. SL=Single Language, ML=Multi-Lingual.

subtask	ST	MT
10-1	80.68	78.96
10-2	52.09	55.57
10-3	36.28	35.16

Table 8: Macro-F1 scores for subtasks of task 10

and much can be gained with a CRF layer. On the contrary, for the second subtask, the difference is negligible, probably due to the short length (and thus easier boundary detection) of the spans.

#### 4.7 Task 9: Multilingual Tweet Intimacy Analysis

The task is to identify intimacy on tweets on a scale of 1-5. More details about the data collection can be found in Pei and Jurgens (2020) and Pei et al. (2022). Besides the six languages available in training (Table 7), Hindi, Arabic, Dutch, and Korean are test-only languages.

We implement the task as a regression task in the MaChAmp toolkit, which uses an MSE loss. We compare mono-lingual models against multi-lingual models and evaluate the use of a shared decoder and language-specific decoders. We use the average (absolute) distance between the gold and predicted label for model selection (because it is the only regression metric currently available in MaChAmp), and report Pearson  $r$  following the official metric.

Results (Table 7) show that each of the model varieties performs best for some of the languages. The highest average is obtained by the multilingual model with a shared decoder, however, it performs bad on French. A manual inspection revealed that this difference is mostly due to bad performance on short sentences.

#### 4.8 Task 10: Towards Explainable Detection of Online Sexism

This task concerns the detection and classification of sexism against women on social media data from Gab and Reddit (Kirk et al., 2023a,b). The task is divided into three subtasks: 1) binary classification (sexist or not), 2) only for sexist posts: in which category does it belong: threats, derogation, animosity, or prejudiced discussions. 3) which subcategory does the post belong to, there are two or three subcategories per main category from subtask 2.

We implement each of these tasks as a classification task and compare it to a model that includes

11	Multi-sep-clas	Multi-sep-reg	Multi-shared-clas	Multi-shared-reg	Single-clas	Single-reg
ArMIS	180.51	42.85	45.32	64.93	108.49	30.14
ConvAbuse	45.44	19.10	30.77	10.78	37.01	12.26
HS-Brexit	10.65	21.48	22.20	9.93	3.26	8.52
MD-Agreement	30.80	25.23	35.16	19.54	41.79	20.43
Avg.	66.85	27.17	33.36	26.29	47.64	17.84

Table 9: Results (cross-entropy) of task11.

lang	SL	ML-shared	ML-sep
am	1.40	0.39	0.17
dz	4.51	13.88	14.25
ha	28.62	27.66	28.03
ig	28.02	25.96	26.77
kr	20.23	11.99	16.53
ma	74.81	73.73	72.05
pcm	36.57	21.77	19.27
pt	9.43	10.06	10.61
sw	9.34	10.02	13.16
ts	29.40	20.99	26.91
twi	30.59	20.66	23.44
yo	27.63	27.03	26.24
Avg.	25.05	22.01	23.12

Table 10: Macro-F1 scores for task 12

all three tasks simultaneously. We use macro-f1 for all tasks, following the official metrics (although task 1 is a binary task).

Results (Table 8) show that the multi-task model is only beneficial for the coarse categories classification (subtask 2). This might be due to this task being in between both other tasks, and thus more closely related to the others. It should be noted that across the tasks there is no error propagation, as subtask 2 is only evaluated on sexism posts and subtasks 2 and 3 are implemented as separate tasks.

#### 4.9 Task 11: Learning with Disagreements (Le-Wi-Di)

Task 11 contains a variety of 4 NLP tasks (Leonardelli et al., 2023): misogyny detection on Arabic tweets (ArMIS), abuse detection in dialogues (ConvAbuse), hate speech detection on data concerning the Brexit (HS-Brexit) and offensiveness detection on tweets from 5 different topics (MD-Agreement). All of these tasks contain soft labels, which are the average scores over multiple annotators (i.e. a float between 0.0 and 1.0). The official evaluation metric is cross-entropy, but we use MaChAmp’s default metrics for each task-type for model selection.

We predict the label as a regression task as well as a classification task, as the number of annotators is between 4 and 10, and the number of labels is

relatively small. Furthermore, we evaluate using a multi-dataset model with a shared encoder as well as separate decoders for each task.

Results (Table 9) show that the regression task type performs better for this task. Single-dataset models outperform the multi-dataset setup, probably because the tasks are too diverse.

#### 4.10 Task 12: AfriSenti-SemEval: Sentiment Analysis for Low-resource African Languages using Twitter Dataset

Officially there are three sub-tasks, but they all concern the same task, they merely differ in the setup. The task is binary sentiment analysis (Muhammad et al., 2023a); the first subtask is for mono-lingual models, the second for multilingual models and the third is for transferring to new unseen languages (Muhammad et al., 2023b). We consider all of these settings and use the best models from the multilingual setting for the cross-lingual setup.

We compare single language models to multilingual models, where we test with shared decoder-heads and separate decoder-heads. Table 10 shows that the single-task models perform best on average. Although, for challenging languages (SL < 10.0) multilingual learning is beneficial.

## 5 Results

The scores on the development data for all our setups (Figure 3) show that the single-task intermediate training is most successful (highest for 12/21). It should be noted, however, that we only plot the best single transfer task, which does need to be found first. The multi-task setup only performs best for one task (8-1), this is because it is the only model that has to share parameters in the model that is used for prediction. For most tasks, the gains obtained with intermediate training are substantial, but there are five cases where the single model performs best; two of these are the soft labels of task 11.

Results on the test data (Table 11) show that al-

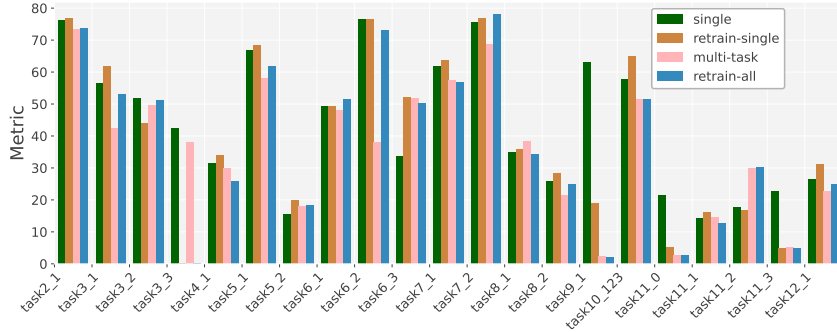


Figure 3: Absolute performance (official metrics) of four settings; the single dataset models, the multi-task models that are trained on all tasks, and the retrained models that are trained on a single (we show the best one only), or on all tasks. For task 11 we use 1/cross\_entropy, so higher is better for all tasks.

Task	Score	Rank	Task	Score	Rank
2	73.74	8/18	8-1	78.40	1/7
3-1	31.67		8-2	40.55	1/6
3-2	38.01		9	57.47	18/46
3-3	29.36		10-1	81.92	53/84
4-1	48	15/42	10-2	62.82	28/69
4-2	34	3/20	10-3	8.04	62/63
4-2	19	10/12	11-1	0.69	15/27
5	65.3	17/25	11-2	1.11	20/27
6	—		11-3	0.47	18/27
7-1	—		11-4	0.61	12/27
7-2	75.6	14/19	12	2.26-51.17	

Table 11: Scores and ranking on test data, official metrics are used. — indicates failed submissions.

though we rank high for 3 tasks (4-2, 8-1, and 8-2), for most tasks, we rank somewhere in the middle. This was expected, as we do almost no tuning per task, no ensembling, use no domain/language-specific embeddings and do not use any non-SemEval data nor data augmentation.

## 6 Analysis

We plot the difference of each retraining setup to the single task baseline in Figure 4; positive numbers mean the baseline was outperformed. Results show that multi-task learning and intermediate training leads to performance loss more often than performance gains in our setup, which is probably a result of having large training sets and good performing baselines. In general, some datasets are particularly good at being on the receiving end of transfer, whereas there is no clear dataset to transfer from (rows are more consistently colored than columns). Tasks that particularly benefit from intermediate training are tasks 6-3, 10, 11-1, and 11-2. For task 6-3, we have a very low baseline, and results seem unstable, task 10 gains from almost all

single task transfers, but mostly from itself (note that we just train twice on the same data here).

The performance of the models trained on all datasets is not more robust compared to the single-task source models, although scaling up further might change this effect (Aribandi et al., 2022). Diverse batching as well as dataset smoothing only lead to performance improvements in specific cases in our setup. We do not find improvements similar to Aghajanyan et al. (2021) for diverse batching, which is probably either because they use the same decoder for all tasks (in a generative model), or because of the scale of datasets.

To evaluate which properties are important for dataset selection in multi-task learning we perform a correlation study. We use the single dataset transfer results (differences to baseline on dev, i.e. first 21 columns of Figure 4) as target variable, and evaluate the correlation against a variety of properties of the tasks. Results (Table 12) show that the strongest (negative) correlations are found for the dataset size variables. For source datasets, we hypothesize that this is because the model overfits towards large datasets, which impedes transfer. For target datasets, this is because the model is undertrained, and there is more space for improvement. Perhaps surprisingly, baseline scores are much less informative compared to dataset size, although baseline scores are expected to correlate with baseline performance. However, it should be noted that different metrics are used, making the correlation less reliable. The number of tasks and number of languages are both negatively related to performance, as there is already sharing going on in the baseline setting, adding even varieties of data is not beneficial. Perhaps surprisingly, task type overlap is also negative, however, this is



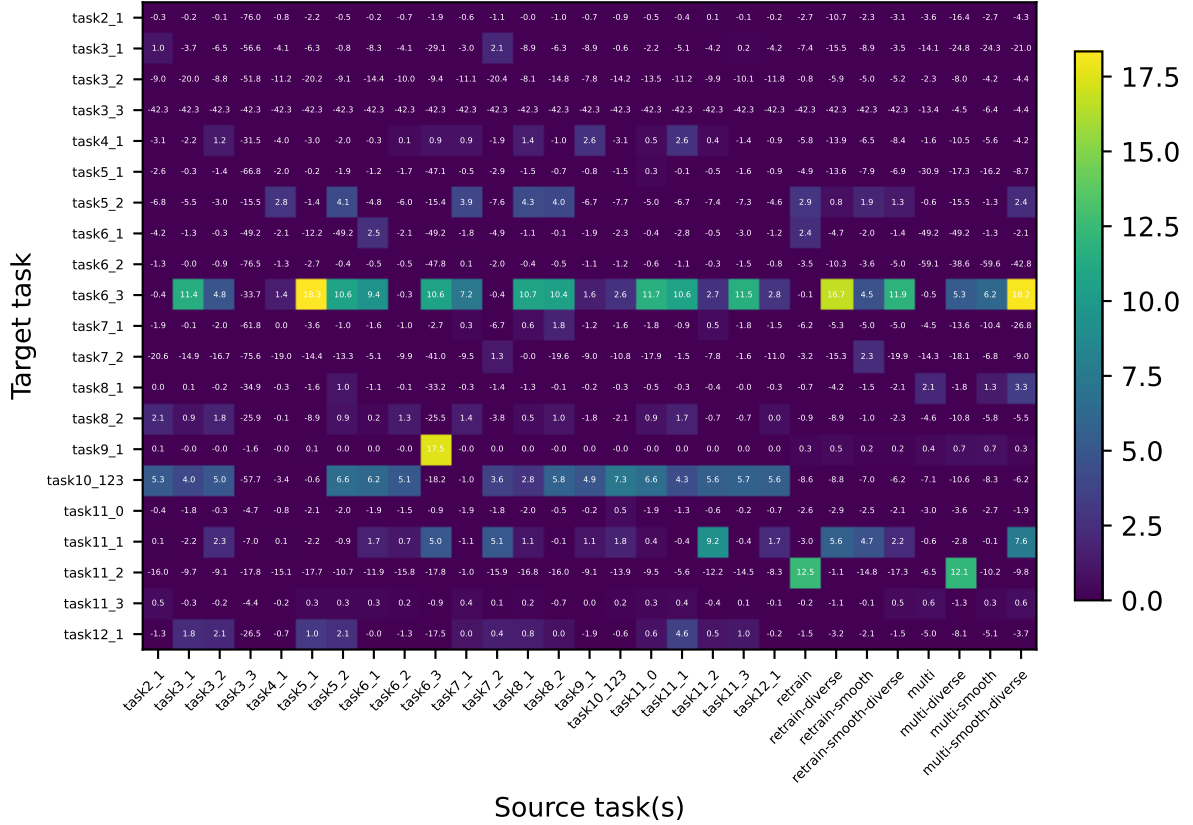


Figure 4: Difference in metric with intermediate training and multi-task learning versus baseline. The target datasets are on the y-axis, the source on the x-axis. Note the the intermediate step can be 1 dataset or all. The last 4 columns represent the multi-task models that are not retrained. Note that the color gradient start at 0 to 30, as we focus on gains. Also note that each row has its own metric, and for the task11 subtasks we use  $(1/\text{cross\_entropy}) * 100$ .

Feature	Description	Pearson
src_base	Baseline performance of src task	0.009
tgt_base	Baseline performance of tgt task	-0.109
base_ratio	src_base/tgt_base	0.093
type_overlap	Overlap in task types	-0.116
num_src_langs	Number of languages in src data	-0.079
num_tgt_langs	Number of languages in tgt data	-0.111
lang_overlap	Overlap in number of languages	0.022
dom_overlap	Whether domains match (binary)	0.088
src_size	# words in src data	-0.500
tgt_size	# words in tgt data	-0.562
size_ratio	src_size/tgt_size	-0.133

Table 12: Pearson correlations between properties of transfer settings and the difference in performance with respect to the baseline.

highly dependent on the choice of datasets, and the task types are imbalanced in our sample.

To gauge the effect of simply upscaling the number of datasets, we also trained models on the combination of the data in Section 4 and the SemEval 2022 datasets used by van der Goot (2022). This leads to approximately twice as large training data, but no robust gains in performance; for some tasks

the combined data increases performance for others it does not (results in Appendix D). Although the scale of data does not match recent studies in the space of generative language models (Chung et al., 2022), our results do suggest that scaling up might not be the most prominent direction for obtaining robust multi-task autoencoder models.

## 7 Conclusion

We evaluate the effect of multi-task learning and intermediate training on all text-based SemEval 2023 tasks and showed that for a non-curated set of benchmarks it is hard to obtain consistent improvements. We found that specific target tasks are likely to gain from intermediate training, but finding the right source dataset/task is non-trivial. Training on a wide variety of tasks has shown to not be more robust compared to finding the best source task to transfer from. In contrast to previous work, having task-heterogeneous batches did not lead to consistent performance gains in our setup.

## Acknowledgements

I would like to thank my colleagues at NLP North and the anonymous reviewers for their feedback on this project. Thanks to all task organizers for providing the data. I acknowledge the IT University of Copenhagen HPC resources made available for conducting the research reported in this paper.

## References

- Armen Aghajanyan, Ankit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. [Muppet: Massive multi-task representations with pre-finetuning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Gupta, Kai Hui, Sebastian Ruder, and Donald Metzler. 2022. [Ext5: Towards extreme multi-task scaling for transfer learning](#). In *International Conference on Learning Representations*.
- Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. [XLM-T: Multilingual language models in Twitter for sentiment analysis and beyond](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 258–266, Marseille, France. European Language Resources Association.
- Ting-Yun Chang and Chi-Jen Lu. 2021. [Rethinking why intermediate-task fine-tuning works](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 706–713, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ethan C. Chau and Noah A. Smith. 2021. [Specializing multilingual language models: An empirical study](#). In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 51–61, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. [Scaling instruction-finetuned language models](#). *arXiv preprint arXiv:2210.11416*.
- Nick Craswell, Daniel Campos, Bhaskar Mitra, Emine Yilmaz, and Bodo Billerbeck. 2020. ORCAS: 20 million clicked query-document pairs for analyzing search. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2983–2989.
- 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.
- Jason Alan Fries, Leon Weber, Natasha Seelam, Gabriel Altay, Debajyoti Datta, Samuele Garda, Myungsun Kang, Ruisi Su, Wojciech Kusa, Samuel Cahyawijaya, et al. 2022. [BigBio: A framework for data-centric biomedical natural language processing](#). In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Maik Fröbe, Tim Gollub, Matthias Hagen, and Martin Potthast. 2023. SemEval-2023 task 5: Clickbait spoiling. In *17th International Workshop on Semantic Evaluation (SemEval-2023)*.
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. [Don’t stop pretraining: Adapt language models to domains and tasks](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.
- Matthias Hagen, Maik Fröbe, Artur Jurk, and Martin Potthast. 2022. [Clickbait spoiling via question answering and passage retrieval](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7025–7036, Dublin, Ireland. Association for Computational Linguistics.
- Mael Jullien, Marco Valentino, Hannah Frost, Paul O’Regan, Donald Landers, and André Freitas. 2023. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data (NLI4CT). In *Proceedings of the 17th International Workshop on Semantic Evaluation*.
- Vivek Khetan, Somin Wadhwa, Byron Wallace, and Silvio Amir. 2023. Semeval-2023 task 8: Causal medical claim identification and related pio frame extraction from social media posts. In *Proceedings of the 17th International Workshop on Semantic Evaluation*, Toronto, Canada. Association for Computational Linguistics.
- Johannes Kiesel, Milad Alshomary, Nicolas Handke, Xiaoni Cai, Henning Wachsmuth, and Benno Stein. 2022. [Identifying the human values behind arguments](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4459–4471, Dublin, Ireland. Association for Computational Linguistics.
- Johannes Kiesel, Milad Alshomary, Nailia Mirzakhmedova, Maximilian Heinrich, Nicolas Handke, Henning Wachsmuth, and Benno Stein. 2023.

- Semeval-2023 task 4: Valueeval: Identification of human values behind arguments. In *Proceedings of the 17th International Workshop on Semantic Evaluation*, Toronto, Canada. Association for Computational Linguistics.
- Hannah Rose Kirk, Wenjie Yin, Bertie Vidgen, and Paul Röttger. 2023a. [SemEval-2023 Task 10: Explainable Detection of Online Sexism](#). In *Proceedings of the 17th International Workshop on Semantic Evaluation*, Toronto, Canada. Association for Computational Linguistics.
- Hannah Rose Kirk, Wenjie Yin, Bertie Vidgen, and Paul Röttger. 2023b. [SemEval-2023 Task 10: Explainable Detection of Online Sexism](#). In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*. Association for Computational Linguistics.
- Elisa Leonardelli, Gavin Abercrombie, Dina Almanea, Valerio Basile, Tommaso Fornaciari, Barbara Plank, Massimo Poesio, Verena Rieser, and Alexandra Uma. 2023. SemEval-2023 Task 11: Learning With Disagreements (LeWiDi). In *Proceedings of the 17th International Workshop on Semantic Evaluation*, Toronto, Canada. Association for Computational Linguistics.
- Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022. [MultiCoNER: A large-scale multilingual dataset for complex named entity recognition](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3798–3809, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Abinew Ali Ayele, Nedjma Ousidhoum, David Ifeoluwa Adelani, Seid Muhie Yimam, Ibrahim Sa’id Ahmad, Meriem Beloucif, Saif Mohammad, Sebastian Ruder, Oumaima Hourrane, Pavel Brazdil, Felermimo Dário Mário António Ali, Davis Davis, Salomey Osei, Bello Shehu Bello, Falalu Ibrahim, Tajudeen Gwadabe, Samuel Rutunda, Tadesse Belay, Wendimu Baye Messelle, Hailu Beshada Balcha, Sisay Adugna Chala, Hagos Tesfahun Gebremichael, Bernard Opoku, and Steven Arthur. 2023a. [AfriSenti: A Twitter Sentiment Analysis Benchmark for African Languages](#).
- Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Seid Muhie Yimam, David Ifeoluwa Adelani, Ibrahim Sa’id Ahmad, Nedjma Ousidhoum, Abinew Ali Ayele, Saif M. Mohammad, Meriem Beloucif, and Sebastian Ruder. 2023b. SemEval-2023 Task 12: Sentiment Analysis for African Languages (AfriSenti-SemEval). In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*. Association for Computational Linguistics.
- Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamel Seddah. 2021. [When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.
- Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.
- 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.
- Mihir Parmar, Swaroop Mishra, Mirali Purohit, Man Luo, Murad Mohammad, and Chitta Baral. 2022. [In-BoXBART: Get instructions into biomedical multi-task learning](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 112–128, Seattle, United States. Association for Computational Linguistics.
- Jiaxin Pei and David Jurgens. 2020. [Quantifying intimacy in language](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5307–5326, Online. Association for Computational Linguistics.
- Jiaxin Pei, Vítor Silva, Maarten Bos, Yozon Liu, Leonardo Neves, David Jurgens, and Francesco Barbieri. 2022. [SemEval 2023 task 9: Multilingual tweet intimacy analysis](#). *arXiv preprint arXiv:2210.01108*.
- Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. [Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks](#). *arXiv preprint arXiv:1811.01088*.
- Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. SemEval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multi-lingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation, SemEval 2023*, Toronto, Canada.
- 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.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. [Multi-task prompted training enables zero-shot task generalization](#). In *International Conference on Learning Representations*.

Rob van der Goot. 2022. [MaChAmp at SemEval-2022 tasks 2, 3, 4, 6, 10, 11, and 12: Multi-task multi-lingual learning for a pre-selected set of semantic datasets](#). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1695–1703, Seattle, United States. Association for Computational Linguistics.

Somin Wadhwa, Vivek Khetan, Silvio Amir, and Byron Wallace. 2023. [RedHOT: A corpus of annotated medical questions, experiences, and claims on social media](#). In *European Association of Computational Linguistics (EACL)*.

Alex Wang, Jan Hula, Patrick Xia, Raghavendra Pappagari, R. Thomas McCoy, Roma Patel, Najoung Kim, Ian Tenney, Yinghui Huang, Katherin Yu, Shuning Jin, Berlin Chen, Benjamin Van Durme, Edouard Grave, Ellie Pavlick, and Samuel R. Bowman. 2019. [Can you tell me how to get past sesame street? sentence-level pretraining beyond language modeling](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4465–4476, Florence, Italy. Association for Computational Linguistics.

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.



### A Task 3: Results

Due to space constraints, results on task 3 can be found here (Table 13).

### B Task 6: LegalEval: Understanding Legal Texts

Task 6 has three subtasks, and consists of legal texts. The first subtask is to identify the rhetorical roles in a text, these can be extracted as spans, and we model them as BIO-labels on the word level. It should be noted that the labels are roughly assigned on the sentence level, meaning that the spans with the same label are often subsequent (i.e. a paragraph that is an *analysis*, can be 5 sentences long, and consist of 5 spans), however, finding the boundaries is also part of the task. We included the None label, because it is also included in the evaluation (and data) as a class. The second subtask is named entity recognition, containing 13 different types of entities. For this subtask, two different types of texts are used: preambles and judgements. The third task concerns predicting the outcome of a legal judgement document (binary: accept or reject), and providing the relevant text spans in the document that explain this outcome.

For subtask 1, we compare the use of a standard sequence labeling to an approach with a CRF layer. For the second subtask, we compare the same types of models, but also use different strategies to handle the multi-dataset setting. We train models on the separate datasets, and attempt to combine them within single decoder, as well as a setup where each dataset has its own decoder. For the third subtask, we could not identify the annotation of the relevant text spans in the provided data, so we only predict outcome as a binary classification.

For subtask 1, it is clear that the CRF layer with BIO constraints is beneficial (Table 14). For subtask 2, the single dataset models perform best and surprisingly, the CRF layer is not beneficial for one of the datasets, leading to a lower average compared to using a simple feedforward layer. For subtask 3, our performance matches the majority vote baseline, this is because the model predicts every case as being rejected. We find two possible reasons for this: 1) the dev data is balanced, but the training data has more rejected cases. 2) the input texts are long (average of 4012 words), and only the first 128 subwords are taken into account in the default settings of MaChAmp (to save memory). We leave a more detailed analysis for future work.

### C Language models comparison

In table 15 we compare the performance of a variety of language models for the best single task settings. For our final test split submissions, we used the best setting of Figure 4, and retrained it with the best language model for each dataset.

### D Training on SemEval 2022 and 2023 data

We use the best settings of [van der Goot \(2022\)](#) for the 2022 data, and train the multi-task model on the combination of all 2022 and 2023 datasets. We plot the best setting (smoothing/heterogeneous batching) for each setup in Figure 5.

<b>3-1</b>	MT-clas- shared	MT-clas- sep	ST-clas						
it	60.56	37.96	35.12						
en	41.85	30.42	35.34						
ru	53.08	51.05	43.96						
po	65.50	42.67	56.94						
fr	65.87	56.24	60.87						
ge	78.57	79.41	63.99						
Avg.	60.91	49.62	49.37						
<b>3-2</b>	ST-clas	ST- multiclas	ST- sep_clas	MT-clas- shared	MT-clas- sep	MT-multi clas- shared	MT-multi clas-sep	MT-sep_ clas- shared	MT-sep_ clas-sep
it	28.96	47.01	26.57	34.21	39.62	32.11	36.36	27.51	37.65
en	57.45	65.24	39.93	57.09	61.06	39.12	39.82	40.34	39.80
ru	18.18	26.53	14.29	34.48	37.59	29.30	32.77	19.05	21.74
po	18.43	47.74	38.35	47.11	42.31	43.12	47.44	35.34	47.52
fr	24.83	41.61	22.86	36.60	40.22	22.70	32.48	19.89	25.40
ge	48.57	43.75	31.22	33.33	36.16	26.61	38.86	28.28	24.11
Avg.	32.74	45.31	28.87	40.47	42.83	32.16	37.96	28.40	32.70
<b>3-3</b>									
it	0.00	0.00	19.11	0.00	0.00	37.16	28.10	16.74	13.67
en	0.00	0.00	0.00	0.00	0.00	23.22	19.45	0.00	0.35
ru	3.12	5.47	16.89	0.00	0.00	34.84	25.36	13.54	10.31
po	8.92	2.09	7.58	0.00	0.00	22.01	17.46	6.34	5.61
fr	0.00	7.99	12.55	0.00	0.00	30.27	28.63	16.52	14.27
ge	27.67	23.77	21.61	0.00	0.00	33.74	24.04	17.29	17.81
Avg.	6.62	6.55	12.96	0.00	0.00	30.21	23.84	11.74	10.34

Table 13: Macro F1 for subtask 3-1, and micro F1 for subtasks 3-2 and 3-3

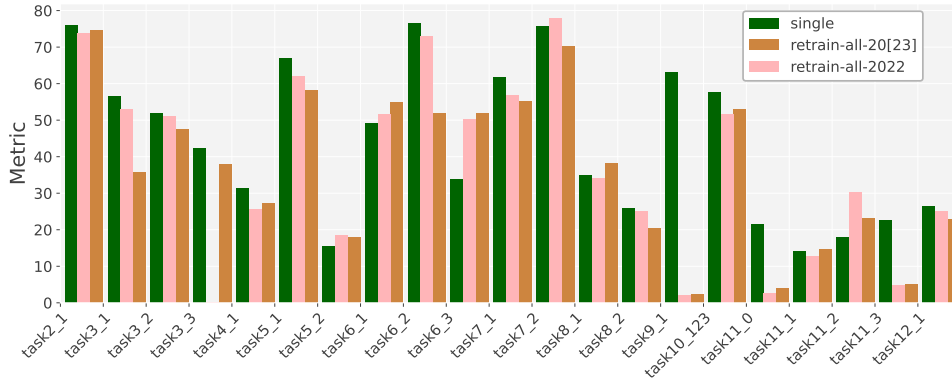


Figure 5: Performance on all tasks when trainin on the SemEval 2023 data only versus training on the combination of 2022 and 2023. Note that performance on task3\_3 is missing due to an error in our scripts; since training the model was computationally expensive we did not redo it.

6-1	seq	bio				
	37.50	49.20				
6-2	ST-seq	ST-bio	MT- shared-seq	MT- sep-seq	MT- shared-bio	MT- sep-bio
Judgement	80.55	82.17	80.44	78.72	80.61	81.02
Preamble	72.23	68.11	71.28	71.22	66.03	67.74
Avg.	76.39	75.14	75.86	74.97	73.32	74.38
6-3	33.00					

Table 14: Results task 6. Span-F1 for subtask 1 and 2, and macro F1 for subtask 3.

	bert-base-multilingual-cased	mdeberta-v3-base	mluke-large	mluke-large-lite	xlm-roberta-large	inforlm-large	mbart-large-50
task2_1_seq-bio_shared	76.03	70.62	72.44	73.28	79.13	79.24	74.58
task3_1_classification_shared	56.56	58.26	27.49	27.49	27.49	27.49	43.64
task3_2_multi-clas_it*	40.79	0.00	0.00	0.00	0.00	0.00	47.66
task3_2_multi-clas_ru*	47.01	0.00	0.00	0.00	29.36	0.00	41.44
task3_2_multi-clas_fr*	51.16	47.55	36.50	36.50	10.17	0.00	32.35
task3_2_multi-clas_ge*	46.03	0.00	0.00	0.00	14.49	0.00	0.00
task3_2_multi-clas_en*	67.96	66.76	64.27	65.84	42.02	42.02	64.26
task3_2_multi-clas_po*	58.04	0.00	0.00	0.00	10.42	0.00	0.00
task3_3_classification_shared	42.26	0.00	0.00	0.00	0.00	0.00	0.00
task4_1_multiclas_comb	31.47	43.84	48.40	47.90	48.31	15.11	42.05
task5_1_classification_1234567	66.80	70.97	36.31	38.74	48.23	31.71	59.32
task5_2_seq-bio_base	15.53	19.74	20.49	20.29	10.28	2.69	20.81
task6_1_seq-bio_mono	49.23	51.03	54.46	55.90	54.54	55.52	52.94
task6_2_seq_NER-TRAIN-JUDGEMENT	80.79	85.00	84.96	86.06	85.66	86.04	83.73
task6_2_seq_NER-TRAIN-PREAMBLE	72.13	87.09	88.78	89.08	88.91	88.04	85.89
task6_3_classification_mono	33.71	34.00	33.79	51.42	44.84	33.27	57.06
task7_1_classification_section	61.82	61.18	58.85	61.58	54.81	50.36	59.36
task7_2_classification_base	75.62	62.59	77.96	66.05	61.85	74.92	69.51
task8_1_seq-bio_base	34.92	36.02	36.12	35.93	36.97	36.65	35.41
task8_2_seq-bio_base	25.90	29.38	28.94	28.62	28.84	24.91	23.62
task9_1_regression_shared	63.06	67.25	70.15	69.15	68.30	67.90	66.05
task10_123_classification_multi	57.74	60.98	62.07	63.85	65.67	56.91	56.76
task11_0_regression_0	21.40	21.66	18.68	22.90	27.66	24.11	20.72
task11_1_regression_0	14.20	23.52	2.21	5.50	7.44	1.94	10.75
task11_2_classification_0	17.83	4.12	6.96	5.82	-0.00	7.28	3.86
task11_3_regression_0	22.70	24.42	26.38	21.08	19.52	25.60	24.76
task12_1_classification_am	0.08	15.22	0.00	0.00	0.00	0.00	3.98
task12_1_classification_dz	8.13	19.11	11.50	12.05	0.00	0.00	6.38
task12_1_classification_ha	28.48	31.37	30.81	30.60	100.00	100.00	30.30
task12_1_classification_ig	29.01	28.90	29.81	29.82	0.00	0.00	27.61
task12_1_classification_kr	21.05	26.31	19.74	18.98	0.00	0.00	15.49
task12_1_classification_ma	76.71	77.26	78.20	77.79	76.84	77.40	75.75
task12_1_classification_pcm	30.66	29.26	25.27	0.00	0.00	0.00	20.00
task12_1_classification_pt	7.44	11.89	17.90	16.89	0.00	0.00	13.94
task12_1_classification_sw	11.06	16.07	0.00	15.43	100.00	22.30	12.31
task12_1_classification_twi	49.96	49.89	47.99	24.40	100.00	100.00	41.67
task12_1_classification_yo	28.92	30.76	28.15	35.29	28.28	35.12	30.13

Table 15: Comparison of different language models for the best single dataset setting of each task. The official metrics for each task are shown. \* due to an error in the code the threshold for including labels was not set correct (except for bert-base-multilingual-cased), results are thus incomplete.