One Model to Rule them All

Bjerva, Johannes

PART IV

Combining Multitask and Multilingual Learning

CHAPTER 8

One Model to rule them all: Multitask Multilingual Learning

Abstract | Multitask learning and multilingual learning share many similarities, and partially build on the same assumptions. One such assumption is that similarities between tasks, or between languages, can be exploited with beneficial effects. A natural extension of these two separate paradigms is to combine them, in order to take advantage of such similarities across both modalities simultaneously. In this chapter, such a combined paradigm is explored, with the goal of building *One Model to rule them all*. The pilot experiments presented here take a first step in this direction, by looking at Part-of-Speech tagging and dependency relation labelling for a large selection of source languages. We restrict ourselves to looking at three target languages representing some typological variety: Finnish, Italian, and Slovene. Furthermore, we run experiments with a relatively simple model, using simple hard parameter sharing, and multilingual input representations. In spite of this simplicity, promising results are obtained for these three languages. For instance, a model which has not seen a single target language PoS tag performs almost equally to a model trained on target language PoS tagging only.

8.1 Combining Multitask Learning and Multilinguality

While Part I of this thesis focussed on multitask learning (MTL), and Part II focussed on multilingual learning (MLL), we now turn to a combined paradigm. In other words, in this final chapter of the thesis, we both consider several tasks and languages at the same time. Let us first consider why such an approach might be useful. For one, joint multilingual and multitask learning allows for taking advantage of the increasing amount of multilingual corpora with overlapping annotation layers, such as the Universal Dependencies (Nivre et al., 2017), and the Parallel Meaning Bank (Abzianidze et al., 2017).[1] Hence, this approach might significantly reduce *data waste*, as one traditionally only considers a single task–language combination at a time. An additional advantage of this paradigm is that it opens up for simultaneous model transfer between languages and tasks. This essentially allows for applying zero-shot learning, in the sense of predicting labels for an unseen task–language combination while taking advantage of other task–language combinations.[2] This approach has not been the subject of much attention in the field, perhaps due to its reliance on the combination of both MTL and MLL, which have only recently become popular. Another issue is the fact that such combined systems put rather large demands on both access to data (alleviated by the UD project), and access to sufficient computing resources. Although a full exploration of the possibilities of this paradigm is not carried out in this chapter, we do take a first step in this direction. The main aim of this chapter is to provide an answer to the following research question, in order to answer **RQ 5**.

RQ 5a To what extent can a combined MTL/MLL system generate sensible predictions for an unseen task–language combination?

[1] In particular, we are interested in the fact that several languages have annotations within the same theoretical framework, following the same annotation guidelines, rather than a single language having several layers of annotation.

[2] I.e., zero-shot learning in a similar sense to Johnson et al. (2016).

Answering this question is considered as a step in the direction of *One Model to rule them all*. If successful, this will allow for bootstrapping off of more-or-less related languages and tasks, which in turn will be highly useful for both low-resource languages and low-resource tasks.

Related work

For related work on the separate paradigms of multilingual and multitask learning, the reader is referred to Chapter 3. In this chapter, we are concerned with a combined multilingual and multitask learning paradigm. Although little work has been done in multilingual multitask NLP, Yang et al. (2016) make some preliminary experiments in this direction by contrasting the two approaches, experimenting with NER, PoS tagging, and chunking on English, Dutch, and Spanish. Their approach uses hard parameter sharing for certain layers, either between languages, or between tasks. In the case of monotask MLL, they share character embeddings and weights of a character-based RNN, whereas in their monolingual MTL setup, they attach a task-specific conditional random field for each task. Since their MLL setup depends on sharing character-based features, the approach is restricted to relatively related languages, and is not likely to work well for less related ones. Indeed, Yang et al. (2016) apply their method only to the relatively closely-related languages English, Dutch and Spanish. Recent work by Fang and Cohn (2017) exploits bilingual dictionaries in order to obtain cross-lingual embeddings, which are used to train a PoS tagger for a source language, which is then applied to a target language with embeddings in the same space.

This chapter expands on previous work by unifying MTL and MLL in a single system, using hard parameter sharing. This allows for taking advantage of similarities between tasks and between languages simultaneously. Rather than sharing character-level features, we fo-

cus on using multilingual word-level input representations. One advantage of avoiding character-level features in a setup using hard parameter sharing, is that reliance on morphological similarities is reduced, which might otherwise lead to negative transfer when considering distantly related languages. The work presented here therefore differs from Yang et al. (2016) in two main ways: i) our method is not restricted to morphologically similar languages, and is applicable to a large portion of (combinations of) the languages in the world; ii) we aim to combine a vast amount of data sources to generate reasonable predictions for a given unobserved task–language pair. Additionally, our motivations are quite different. Where Yang et al. (2016) aim to improve performance on a task–language pair for which annotated data exists, by adding a distant supervision signal from a different language for the same task, or a different task for the same language, we aim to induce tags for task–language combinations for which no annotated data exists. This is important, since a multitask multilingual setting will usually resemble the scenario depicted in Table 8.1.[3] Let us consider language l_6 and task t_3, as highlighted in red in the table. In order to fill this *gap*, we can choose from a few approaches: i) Spend an enormous effort in finding, hiring and training annotators; ii) Apply annotation projection to the text snippets which happen to be parallel text with languages for which annotation exists for t_3, or first translate the data from l_6 to such a language (cf. the translation approach described in Chapter 3);[4] iii) Train a multilingual system with supervision from only languages with annotation for t_3 (cross-lingual model transfer); or iv) Train a system on several task–language pairs in the matrix, including l_7 for other tasks.[5] Our

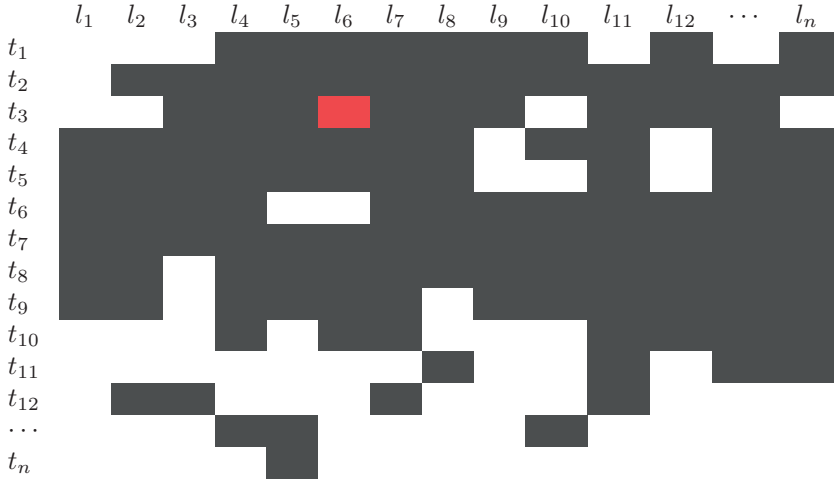
[3]We will refer to such tables as *gap tables*, as they contain some filled (black) cells, and several white *gaps* without data.

[4]Note that the requirements for annotated/parallel data are quite high in this case (cf. Tiedemann et al. (2014)).

[5]The first three approaches can be considered traditional approaches, and are detailed in Chapter 3.

approach is essentially this final approach (iv).

Table 8.1: Black cells indicate the availability of annotated data for a given task–language pair. Some languages have (almost) all cells filled, whereas some have a large amount of gaps. The red cell indicates a potential target task–language combination, for which no annotated data exists.



8.2 Data

8.2.1 Labelled data

While the goal is to extend this approach to a matrix such as in Table 8.1, we will only look at two tasks in this chapter. Additionally, since the goal in this pilot experiment is to see whether the proposed MTL/MLL paradigm is at all feasible, a setting in which very high correlations between main and auxiliary tasks can be found was chosen. We therefore focus on PoS tagging and dependency relation (DepRel) labelling, as data for both of these tasks is available for a relatively large amount of languages through the Universal Dependencies (UD) project. There are many possible ways of defining dependency rela-

tion labels, and in this chapter we use the *simple/simple* paradigm described in Chapter 5 (Table 5.1). Furthermore, positive results have been obtained for this particular task combination, e.g., in Chapter 5. In the experiments of this chapter, UD v1.2 is used (Nivre et al., 2015).

8.2.2 Unlabelled data

We take a similar approach to enabling model multilinguality as in Part III. That is to say, we use unified input representations, in the sense that we use multilingual word embeddings. Although many options exist, we use multilingual skip-gram (Guo et al., 2016), for the same reasons as in Part III.

We train the embeddings in two resource settings. The first is a high resource setting, in which we have access to a large amount of parallel text. In this setting, we train embeddings on the Europarl corpus (Koehn, 2005). The second is a low resource setting, in which we only require very limited amounts of parallel data, and train embeddings on a collection of Bible corpora. The low resource setting is, indeed, truly low-resource, as we only require approximately 140,000 tokens of training data.[6] Additionally, although the use of Bible corpora for multilingual representations can be criticised for many reasons, including the specificity of the domain, and the archaicness of the language, this data has the advantage that it is available for more than 1,000 languages. While this does leave a long tail of approximately 5,000 languages for which such resources are not available, it nonetheless constitutes a leap forward from requiring Europarl-levels of data.

[6]This is the approximate token count for the English New Testament, and is bound to differ for other languages.

8.3 Method

8.3.1 Architecture

The system used is the same bi-GRU as described in Chapter 7. To recap, the bi-GRU is two layers deep, and uses only word-level multilingual embeddings as input. The network has two output layers – one for PoS tags, and one for dependency relation tags. In the settings in which no dependency relations are observed, the weights of the corresponding layer are left unaltered. This includes the baseline setting, and the settings with transfer solely from PoS tags.

Although using character-level representations would likely yield higher performance for some cases, there are two main reasons why this is not done. Primarily, it is likely that using such representations would lead to negative transfer between less related languages. This would need to be dealt with in a more sophisticated way than simple hard parameter sharing, for instance by using sluice networks, in which the parameter sharing itself is learnt (Ruder et al., 2017). Additionally, the goal of the experiments in this chapter is not to obtain the highest possible results, which is the trend in much of current NLP, but rather to investigate the differences between different transfer settings.

8.3.2 Hyperparameters

Hyperparameters were tuned to a small extent on the English development set when training on only English PoS tags, with the goal of asserting that the system performs on-par with the word-based Bi-GRU in Chapter 4. The aim was to perform a relatively low amount of tuning, keeping parameters at fairly standard values. These hyperparameters were used for all experiments in this chapter. We use rectified linear units (ReLUs) for all activation functions (Nair and Hinton, 2010). We apply dropout ($p = 0.2$) at the input level, and recurrent

dropout (Semeniuta et al., 2016) between the layers in the network. We use the Adam optimisation algorithm (Kingma and Ba, 2014) with a batch-size of 10 sentences (randomly sampled from all source languages under consideration). Training is done over a maximum of 50 epochs, using early stopping monitoring the loss on development sets of all source languages in the given experimental condition. The weighting parameter λ, defining the weight of the auxiliary task is set to $\lambda = 1.0$, i.e., weighting the main and auxiliary tasks equally.

8.4 Experiments and Analysis

For the purposes of evaluating whether the proposed approach is feasible, we consider several potential scenarios. In all experiments, we look at filling the *gap* of Finnish, Italian, and Slovene PoS tags. These languages were chosen so as to represent some level of typological diversity, with one language from outside the Indo-European family (Finnish), and two fairly dissimilar Indo-European languages, of which one is a Romance language (Italian), and one is a Slavic language (Slovene). The evaluation metric used in the experiments is the accuracy of PoS tagging on each of these languages, as evaluated on their UD development sets.[7] We will successively increase the amount of, and variety of, data which the models are trained on. We will also investigate the effect of adding more or less related languages to the training data. Language relatedness is displayed in Table 8.2, and is defined heuristically, based on typological relatedness.

Every system is trained on the concatenation of the entire training set of all source languages involved in the setup at hand. Validation is done on the concatenation of the development sets of all source languages in the setup at hand.

[7]No tuning is performed on this set.

Table 8.2: Source language overview table. Columns indicate whether the languages are considered to be related to the header of that column.

Language	Finnish	Italian	Slovene
Basque			
Bulgarian			x
Croatian			x
Czech			x
Danish			
Dutch			
English			
Estonian	x		
French		x	
German			
Hebrew			
Hindi			
Hungarian	x		
Indonesian			
Irish			
Kazakh			
Latin		x	
Greek			
Norwegian			
Persian			
Portuguese		x	
Romanian		x	
Spanish		x	
Swedish			
Tamil			
Turkish			

Training on English PoS

Throughout the experimental overview, we will consider a gapped table as shown in Table 8.3. The black cells denote the source task–language combinations, and the red cells denote the target task–language combinations. Note that, although we could train a joint system for all target languages, separate systems are trained for each target language. This is because that, in following experiments, we look at languages which are related to the target languages to a smaller or larger extent. As our target languages are typologically quite different, this requires us to train separate systems, as it would otherwise be impossible to add a language which is equally related to, e.g., both Finnish and Italian. In the first experiment, we train on English PoS tags only, and evaluate on Finnish, Italian, and Slovene (Table 8.3). We also train a monolingual baseline system for each of the target languages which is used throughout this chapter, with the same general setup as the other systems, using the high-resource multilingual embeddings.

Table 8.3: Gap table – Training on English, PoS only. Evaluation is on the target languages Finnish, Italian, and Slovene.

Language	PoS	DepRel
English		
Target languages		

The results from this setting can be observed in Figure 8.1. The red bars indicate the systems trained on English PoS, with a black border around the system using high resource embeddings, and no border for the system using low resource embeddings. The black bars indicate the monolingual baseline systems. Not surprisingly, transfer from English is not particularly successful, with performance far below baseline. This shows that training on a single relatively unrelated language is not sufficient in this setting. As expected, results

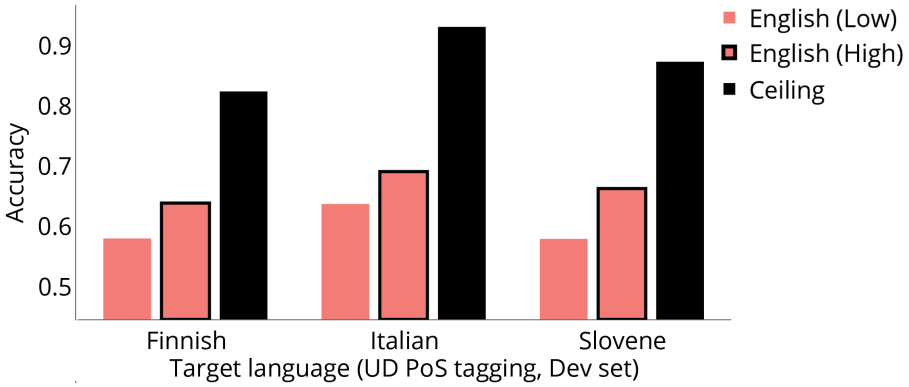


Figure 8.1: Red bars indicate training on English PoS. No border indicates training with low resource embeddings, and a black border indicates training with high resource embeddings. The black bars denote the monolingual baselines.

when using embeddings trained on Europarl are somewhat higher than when using low resource embeddings.

Training on PoS from several languages

Next, we add PoS training data from several languages, all relatively unrelated to the target languages. In this setting, the system for each target language is trained on all languages labelled as unrelated to the source language in Table 8.2, as depicted in Table 8.4.

Table 8.4: Gap table – Training on unrelated languages, PoS only.

Language	PoS	DepRel
Unrelated languages		
Target languages		

The results can be seen in Figure 8.2. Adding more languages to

the training material does not affect results noticeably for Finnish. For Italian and Slovene, however, the results improve somewhat, most notably when using high resource embeddings. This might be due to the fact that the UD dataset contains an Indo-European bias, meaning that the so-called unrelated languages which we have added still share fairly distant ancestry. We also see a rather large increase in the performance on Italian as compared to Slovene. This can be explained by the fact that many of the unrelated languages added in this setting are Germanic. Morphological complexity of Germanic languages is arguably relatively similar to Romance languages, such as Italian. On the other hand, Slavic languages such as Slovene are much more morphologically complex. This might have an effect on the quality of the multilingual word embeddings, leading to training on Germanic languages being more beneficial for Italian than it is for Slovene. Finnish, being from the Finno-Ugric language branch, does not benefit from this setting, perhaps due to its typological distance from the added languages being larger.

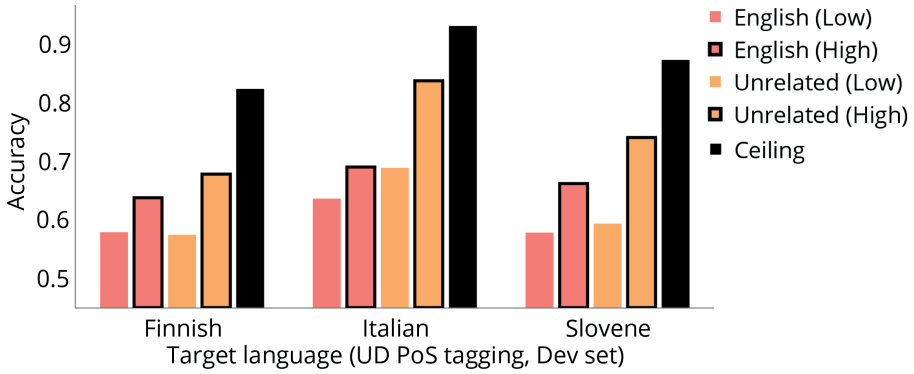


Figure 8.2: Red bars indicate training on English PoS. Orange bars indicate adding unrelated language PoS. No border indicates training with low resource embeddings, and a black border indicates training with high resource embeddings. The black bars denote the monolingual baselines.

Adding source language dependency relations

We now add dependency relation data for the same collection of languages as in the previous setting (see Table 8.5). This is the first setting in which MTL is combined with the MLL experiments, as the network is now trained on both PoS tagging and dependency relation labelling. The idea is that the correlations between PoS tags and DepRel labels can be learnt by the network in an implicit manner, which might be beneficial for system performance. However, we do not expect positive results in this particular setting, considering that the mutual information between PoS tags and dependency relations is relatively high, and we are not adding any extra data (see Chapter 5).

Table 8.5: Gap table – Training on unrelated languages, PoS and dependency relations.

Language	PoS	DepRel
Unrelated languages		
Target languages		

Figure 8.3 shows that, indeed, this addition does not affect results to a large extent. In fact, results drop somewhat in most settings, which may be owed to the fact that some of the net capacity is wasted, since two tasks need to be learned. As expected, since the system has not seen any data for either task for the target languages, adding this data does not improve much, which can be explained by the findings in Chapter 5.

Adding target language dependency relations

In this experiment, we add dependency relation data for the target languages in training (Table 8.6). The intuition behind this, is that the neural network ought to be able to make use of the implicitly

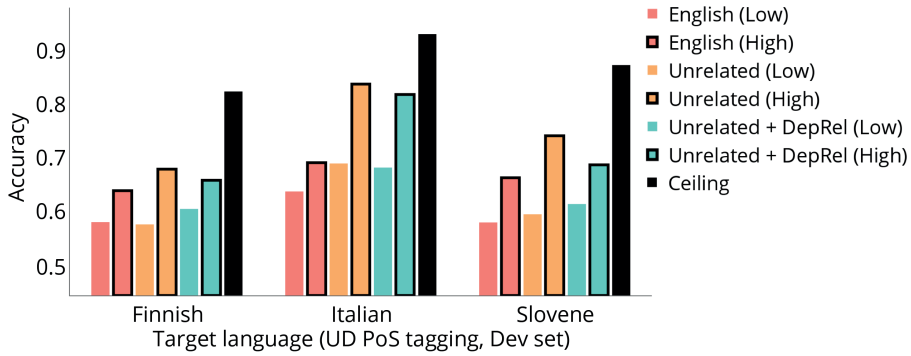


Figure 8.3: Green bars indicate adding source language dependency relations. No border indicates training with low resource embeddings, and a black border indicates training with high resource embeddings.

learn correlations between tasks, thus learning to produce sensible PoS tags for the target languages, in spite of never having actually observed such tags. In a sense, this is the first real combined MTL/MLL experiment in this chapter.

Table 8.6: Gap table – Training on English and unrelated languages, PoS and dependency relations.

Language	PoS	DepRel
Unrelated languages		
Target languages		

Results in Figure 8.4 show high resource embeddings almost reaching ceiling performance. This can be interpreted as showing that the network has learned the correlations between the two tasks, allowing for generating sensible PoS tags for the target languages. Another potential explanation is that adding extra data with high mutual in-

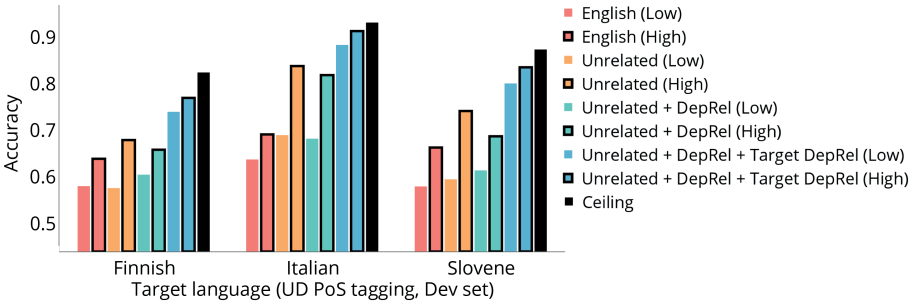


Figure 8.4: Blue bars indicate adding target language dependency relations. No border indicates training with low resource embeddings, and a black border indicates training with high resource embeddings.

formation with PoS tagging ought to be useful, based on the findings in Chapter 5.

Adding related languages, no target dependency relations

We here add data for related languages, as defined in Table 8.2. This is the same as the third experimental setting (Table 8.5), except we also look at related languages (depicted in Table 8.7). That is to say, we do not see any dependency relation tags for the target languages in this setting.

Table 8.7: Gap table – Training on unrelated and related languages, PoS and dependency relations.

Language	PoS	DepRel
Unrelated languages		
Related languages		
Target languages		

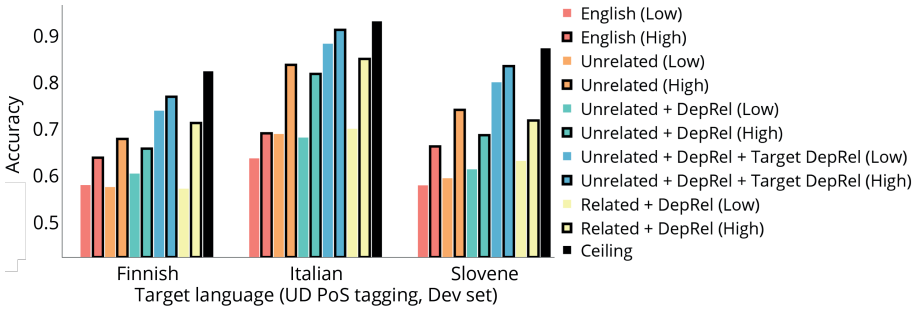


Figure 8.5: Green bars indicate adding source language dependency relations. Yellow bars indicate training on related languages. No border indicates training with low resource embeddings, and a black border indicates training with high resource embeddings.

Comparing the yellow and green bars in Figure 8.5, the change in results is not as large as what might have been anticipated. This is somewhat surprising, as one could expect adding related languages to the mix to improve results significantly. Nonetheless, especially in the high resource setting, some gains can be observed. These results support the findings of Chapter 7, in that training on similar languages can be beneficial in a multilingual scenario. The relative gain for Finnish is especially high, which can be explained by the fact that the model finally has access to source data which to some extent resembles the target data. As for Slovene and Italian, the gains in getting access to Slavic and Romance data, respectively, does not provide a very large benefit as compared to having access to Indo-European data. As for the low resource settings in this experiment, very small differences can be observed. This hints at the possibility that, without access to any target language data, the Bible-based embeddings are close to a performance ceiling.

Adding related languages, with target dependency relations

Finally, we also add in the dependency relation data for the target languages (Table 8.8). This denotes the most complete experimental setting, as we only have a single gap to fill in the table, and have access to the largest possible amount of data.

Table 8.8: Gap table – Training on unrelated and related languages, PoS and dependency relations, as well as target language dependency relations.

Language	PoS	DepRel
Unrelated languages		
Related languages		
Target languages		

The results for this experiment are positive for all three languages, showing that it is possible to output PoS tags of decent quality, without having seen a single target-language PoS tag (Figure 8.6). Performance on Italian is especially promising, reaching the same level as the ceiling baseline, while Finnish and Slovene also show positive results.[8] Notably, although we use training data from related languages, the change in performance between this setting (purple bars) and the corresponding setting with unrelated languages (blue bars) is quite small. Also noteworthy is the small distance between the two embedding types in this setting. While ceiling performance is observed when using high resource embeddings, the low resource scenario also yields positive results. Whereas most of the experiments did not provide much difference in the results with low resource embeddings, the two settings in which we have access to target-language data show that it may be sufficient with this resource scenario. Should

[8] An important caveat, however, is the fact that the setup is rather artificial, as one rarely will have dependency relation annotation for a language, without access to PoS tags. This is discussed further in Section 8.5.

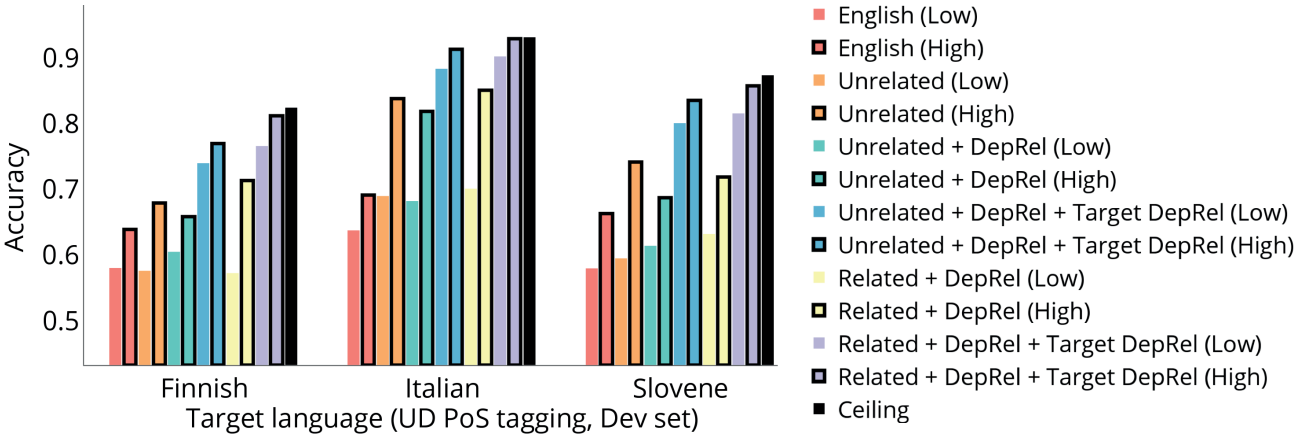


Figure 8.6: Red bars indicate training on English PoS. Orange bars indicate adding unrelated language PoS. Green bars indicate adding source language dependency relations. Blue bars indicate adding target language dependency relations. Yellow bars indicate training on related languages. Purple bars indicate adding target language dependency relations. No border indicates training with low resource embeddings, and a black border indicates training with high resource embeddings. The black bars denote the monolingual baselines.

these results generalise to more exotic languages than the ones used in this study, then this type of multitask multilingual learning might indeed be a useful step towards improving NLP for low-resource languages.

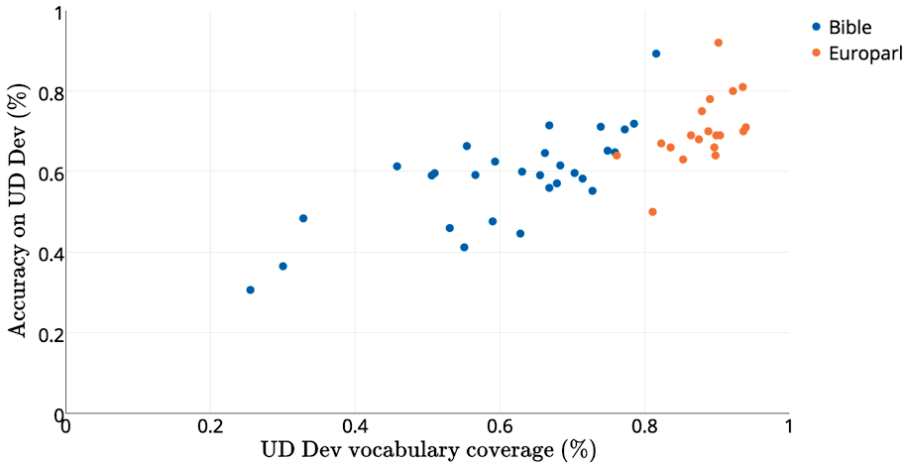


Figure 8.7: Accuracy of monolingual models on UD Dev compared to the vocabulary coverage of the high and low resource embeddings.

8.5 Discussion

While the general results in the high-resource scenario were positive, and indicate that the approach taken here is, at the very least, methodologically sound, the results were more varied in the low-resource scenario. In general, the Bible-based embeddings did not yield any positive results, save for the experiments in which we also had access to some target-language training data. This might be explained further by taking a look at model performance as compared to vocabulary coverage. Figure 8.7 shows the accuracy of monolingual models on UD Dev compared to the vocabulary coverage of the high resource and low resource embeddings on the same UD Dev

set. These models are trained in the same way as the ceiling baseline from the experimental setup in this chapter. There is a relatively strong correlation between accuracy and vocabulary coverage, with most of the low-coverage region naturally occupied by the low-resource embeddings. The fact that the link with vocabulary coverage is as pronounced as what we observe explains the large increases we observe when adding target language training data. In doing so, we effectively increase the portion of the target language vocabulary which the model has observed, thus increasing performance on target language PoS tags.

While the results of the experiments in this chapter seem promising, there are some points which can be criticised. One such matter, is the fact that it is hard to imagine a situation which is exactly as what was described here. The assumption of the experiments was that we did have access to dependency relation annotations for the target languages, but did not have PoS tagged data. As dependency relations constitute a more detailed level of description, this scenario is most likely not a very common one. An interesting direction would therefore be to invert this setting, by trying to fill a dependency relation gap. This is likely much more challenging than the current setup, as the mapping from PoS tags to dependency relations is more heterogenous than the inverse. However, even though filling a gap for more intricate annotations than one has for a language is an interesting problem, filling a gap with annotations at a similar level as what already exists is also a useful application. For instance, some languages have their own PoS tagged corpora, while they do not have any UD annotations. This might be solved by mapping from one PoS tag set to another with the approach described in this chapter.

In spite of the aforementioned issues, the aim of the pilot study is to investigate whether or not the proposed combination of MTL and MLL is at all feasible. The fact that we have seen positive results in such a simplistic setting, using hard parameter sharing and multilin-

gual input representations, certainly indicates that this is the case. A potential approach for dealing with languages for which parallel text does not exist in sufficient quantities, is to rely on bilingual dictionaries instead, as done by Fang and Cohn (2017).

Refining this approach in the future therefore constitutes a highly interesting research direction, for which some approaches are detailed in the next and final chapter of this thesis, in Section 9.4.

8.6 Conclusion

We attempted to combine the paradigms of multilingual and multitask learning. Providing the model with data for the target task for source languages, as well as auxiliary task data for the target and source languages, yielded promising results. In fact, the results are almost on par with training a system directly on the target/source language, indicating that combining the paradigms of MTL and MLL has potential (**RQ 5a**). Although the experimental setup was somewhat artificial, as we assumed access to a more complex level of annotation than what we aimed at producing, this approach constitutes a research direction which is worthwhile pursuing in the future.