Towards Combining Multitask and Multilingual Learning*

Matus Pikuliak, Marian Simko, and Maria Bielikova

Slovak University of Technology in Bratislava, Ilkovicova 2, Bratislava, Slovakia

Abstract. Machine learning is an increasingly important approach to Natural Language Processing. Most languages however do not possess enough data to fully utilize it. When dealing with such languages it is important to use as much auxiliary data as possible. In this work we propose a combination of multitask and multilingual learning. When learning new task we use data from other tasks and other languages at the same time. We evaluate our approach with neural network based model that can solve two tasks – Part-of-speech tagging and Named entity recognition – with four different languages at the same time. Parameters of this model are partially shared across all data and partially they are specific for individual tasks and/or languages. Preliminary experiments show that this approach has its merits as we were able to beat baseline solutions that do not combine data from all the available sources.

Keywords: Transfer Learning · Multilingual Learning · Deep Natural Language Processing.

1 Introduction

Modern machine learning approaches to natural language processing (NLP) are notoriously data hungry. Currently there is a significant disparity in volume of available datasets between various languages. While English, Chinese and a handful of other major languages have the most data, other languages are seriously lacking. This is naturally slowing down the research and development of crucial NLP algorithms, models and services for these low resource languages.

Collecting new data is laborious and expensive. Transfer learning is sometimes proposed as a possible remedy. Instead of creating new datasets we might try to utilize existing ones, even though they are not completely related to our problem. In NLP this means using data from other tasks, languages or domains. Research so far predominately focused on only one of these options at the time. The novelty of our work lies in the fact that we are combining multitask and multilingual learning.

We combine them to utilize as much available data during learning as possible. We believe that combining data from multiple sources might be crucial when trying to create robust and efficient solutions. This is especially important for low-resource languages as it might significantly reduce data requirements.

* Supported by organization x.

We propose a method of training a model with multiple tasks from multiple languages at the same time. This model is theoretically capable of handling variety of tasks, so far we experimented only with two: part-of-speech tagging (POS) and named entity recognition (NER). Both these tasks were solved for four languages (English, German, Spanish, Czech). We evaluated the performance of this model when trained on target data only versus when trained on all available data. In some cases we noted significant score improvements.

2 Related Work

Parameter Sharing. Parameter sharing is a popular technique of multitask learning. Multiple models trained on different tasks share the values of subset of parameters. Such sharing can boost the results for any of the relevant tasks by the virtue of having additional data influencing the training process. Various combination of tasks were already considered in the NLP [19, 23, 16, 13]. [20] use parameter sharing with multiple tasks to create robust sentence representations.

Multilingual Learning. Multilingual learning can be perceived as a special type of multitask learning, when the samples come from different languages. The goal is to transfer knowledge from one language to others. Various techniques to multilingual learning exist, using annotation projection [5, 1], language independent representations [24, 2] or parameter sharing [23, 7]. Parameter sharing techniques are the most relevant to us. They share certain layers between more languages and these layers are therefore becoming multilingual. We extend this idea and combine multilingual learning with multitask learning. To the best of our knowledge this was not done before with parameter sharing. We are aware of two works that project annotations for both dependency parsing and POS tagging at the same time [1, 21].

3 Proposed Method

We propose a multitask multilingual learning method based on parameter sharing. In our approach, the training is done with multiple datasets at the same time. In this work we work with two tasks – NER and POS – and four natural languages. In effect we have 8 unique training datasets and for each of this datasets we have a separate model that is being trained. All these models have the exact same neural network based architecture. In various experiments selected parameters are shared between certain models to achieve transfer of information between them. Sharing parameters in this case means have identical weight matrices on selected layers of a neural network.

3.1 Architecture

Architecture of our model needs to be general enough to allow us to effectively solve multiple tasks at once. In our work we use sentence level tasks, i.e. we

expect a sentence as an input to our model. What is the output depends on the task. Our model is not suited to process higher level units, such as paragraphs or documents. We propose a model with three consecutive parts:

Part 1: Word embeddings. Word embeddings are fixed length vector representation of words that became very popular in previous years [8, 14] as they are able to significantly outperform other types of word representations, namely one-hot encodings. They are based on an idea of using language modeling as auxiliary task when learning about words. The first step of our architecture is to project the words into vector space. For each language L we have a dictionary of words D_L and a function d_L that takes given word and returns an integer identifier of this word in given dictionary. $d_cs(on)$ is a id of the word *on* in Czech while $d_en(on)$ is its id in English. Even though their form is the same, their id might be similar. Word with same id from different languages do not have any connection between them.

For each language we also have an embedding matrix E_L whose i-th row is a word representation for a word W for which $d_L(W) = i$. The matrix E_L has dimensions $a \times b$, where a is the number of words in dictionary D_L and b is the arbitrarily set length of word representation that is set before the word embeddings are created. For words that are not in the dictionary of its language we use zero vector.

If the input of our model is a sentence of N words from language L:

$$I = \langle W_1, W_2, W_3, ..., W_N \rangle \tag{1}$$

For clarity we define a function d' that return a vector for given word as:

$$d'_L(W) = E_L(d_L(W)) \tag{2}$$

With this we can then define the output of this model layer as a sequence of embeddings:

$$e = \langle d'_L(W_1), d'_L(W_2), , d'_L(W_3), , ..., d'_L(W_K), \rangle \tag{3}$$

In our case we use so called multilingual word embeddings as pre-trained word representations that are stored in the E matrices. Multilingual word embeddings are an extension of standard word embeddings technique where words from multiple languages share one semantic vector space [17]. Semantically similar words have similar representations even when they come from different languages. This is the only information that explicitly tells our model what is the relation between various languages and their words. Sometimes researchers let the word embeddings be trainable parameters. In our case we fix them so we do not lose this link between languages.

Part 2: LSTM. Word embeddings are processed in bi-directional LSTM recurrent layer [9]. The weights of this layer are shared across both tasks and languages – the same LSTM layer is used during each pass in the network. This is the part that contains majority of trainable parameters and is therefore also responsible for most of the computation that gets done. This is also where

most of the information sharing happens. The output of this layer is sequence of contextualized word representations. While in the previous layer the same word always have the same representations, here the same word will have different representation if it is used in different sentences.

We used LSTM recurrent layer as they are able to partially tackle the forgetting problem of basic recurrent networks, which tend to forget what they saw in previous steps if the input sequence length is too big. LSTM's use several gates that let the model learn what parts of signal should it keep between the steps and which part should be forgotten. The LSTM is traditionally defined as follows:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{4}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{5}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{6}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma(W_c x_t + U_x h_{t-1} + b_c) \tag{7}$$

$$h_t = o_t \circ \sigma(c_t) \tag{8}$$

where x_t, h_t and c_t is the LSTM input, output and cell state at the time t. The size of h and c can be set arbitrarily. f_t, i_t and o_t are forgetting, input and output activation gates that govern how much signal is kept during the computation. W's, U's and b's are trainable weights and biases. Finally $\circ$ is Hadamard product and σ is a non-linear activation function.

This defines an $LSTM$ function that takes a sequence of inputs and returns a sequence of outputs that encode the state of the LSTM at individual timestamps:

$$LSTM(\langle x_1, x_2, x_3, ..., x_K \rangle) = \langle h_1, h_2, h_3, ..., h_K \rangle \tag{9}$$

The bi-directional LSTM layer of our model is then defined with two LSTM networks. First one processes the word embeddings from the start, while the second one processes it from the end:

$$h_1, h_2, h_3, ..., h_K = LSTM(e) \tag{10}$$

$$\langle h'_K, h'_{K-1}, h'_{K-2}, ...h'_1 \rangle = LSTM(reverse(e)) \tag{11}$$

The output q of this layer is finally defined as:

$$q = \langle (h_1; h'_1), (h_2; h'_2), (h_3; h'_3), ..., (h_K; h'_K) \rangle \tag{12}$$

with semicolon marking a concatenation of the two vectors and e still being an output of previous layer.

Part 3: Output layers. Finally the output of LSTM is processed by task-specific layers (architectures might differ depending on the tasks). The parameters of this part might or might not be shared across languages. So far we experimented with two tasks, part-of-speech tagging and named entity recognition. As both of them are sequence tagging tasks we use the same architecture for this part.

Each contextualized word representation from bi-LSTM layer is used to predict the appropriate tag for given word. To use this we use simple linear projection:

$$p = Wh + b \tag{13}$$

where p is the prediction vector containing probabilities for each possible tag within given task, W and b are weights and biases and h is a contextualized vector for one particular word from previous layer. We use the same parameters (W and b) for each word.

All these predictions for one sentence are then processed by CRF sequence modeling algorithm [12] to calculate final results. Using this algorithm means that instead of simply optimizing for the p to predict the correct tag as much as possible we also take into account the order of individual tags. To this end a transition matrix counting how many times one particular tag followed all the other tags are used. During training this step is differentiable but during inference we need to use dynamic programming to calculate the final tags from the predictions p our model generates while also taking the transition matrix into account. Detailed description of this part of the network is outside of the scope of this article, but we refer to the [12] for more thorough explanation.

3.2 Training

We consider several training modes, based on what kind of data is the model exposed to:

1. **Single task, single language (ST-SL).** This is the standard machine learning setting when we have data for one task from one language only.
2. **Multitask (MT).** More tasks from single language are solved at once, e.g. we train both NER and POS for English.
3. **Multilingual (ML).** Data from multiple languages are used to train one shared task, e.g. we train POS on all languages at the same time.
4. **Multitask, multilingual (MT-ML).** Combination of multitask and multilingual learning. Multiple tasks are solved for multiple languages all at the same time.

We use epochs with fixed amount of training steps. When training with more languages and/or tasks, each training step consists of several minibatches – one minibatch for each relevant datasets. E.g. during multitask learning we might have two relevant datasets, English POS and English NER. This means that we first run one English POS minibatch followed by one English NER minibatch as one training step. Minibatches always contain the same amount of samples.

Each model processes $epochs \times steps \times datasets$ minibatches during training. In effect this means that model gets exposed to more data with increasing number of datasets used for training. This is balanced by the fact that it needs to work with several data distributions at the same time. Naturally during each pass only the parameters that are relevant for a given task and language gets updated. Rest of the parameters lie unchanged.

4 Experiments and Results

4.1 Datasets

In our experiments we used four languages (English, German, Spanish and Czech) and two tasks (part-of-speech tagging, named entity recognition). This means that in overall we had 8 datasets, each with training, development and testing part. The amount of annotated sentences for each dataset can be found in Table 1.

Table 1. Number of sentences in datasets (in thousands).

	en	es	de	cs
NER train	38.4	7.1	24.0	7.2
NER dev	4.8	1.6	2.2	0.9
NER test	4.8	1.4	5.1	0.9
POS train	12.5	14.1	13.8	68.5
POS dev	2.0	1.4	0.8	9.3
POS test	2.0	0.4	1.0	10.1

Part-of-speech tagging. For POS we used Universal Dependencies datasets [15] for each language. These are annotated using universal POS tagging schema containing 17 common tags.

Named entity recognition. We used Groningen Meaning Bank [4] for English, GermEval 2014 NER dataset [3] for German, CoNLL 2002 [18] for Spanish and Czech Named Entity Corpus [11] for Czech. The tagging schemata differ between these datasets so we had to unify them ourselves. We converted them to standard BIO schema used for NER. We used 4 types of named entities (persons, locations, organizations and miscellaneous). English dataset was the only one that did not have separated training, development and testing data so we split it with 8:1:1 ratio.

Word embeddings. For multilingual word embeddings we use publicly available MUSE embeddings [6]. They have word vectors of size 300 for 200,000 words in each language. Vectors for other words were set to zero.

4.2 Experiment

We trained our model in all modes as mentioned in Section 3.2. Every time we have 8 models for each task-language combination. In various settings they share different parameter subspaces. When using multilingual learning they share all the parameters (in effect this means they are identical so it is one model being trained with more data). When using multitask learning the two models share LSTM layer, but the task specific weights used to make the final tag predictions are naturally not shared across tasks. When using multitask multilearning models, they still all share the LSTM layer, while the output layer is shared only

Table 2. NER results for various learning modes for individual languages. Results are per tag F1 scores.

	en	es	de	cs
ST-SL	77.3	73.0	73.3	66.2
MT	77.4	74.3	**75.3**	67.8
ML	**78.1**	75.6	74.6	68.1
MT-ML	77.5	**77.1**	74.3	**69.8**

Table 3. POS results for various learning modes for individual languages. Results are per tag accuracy scores.

	en	es	de	cs
ST-SL	90.66	94.16	91.19	**94.06**
MT	90.90	94.19	91.27	94.05
ML	91.17	94.30	**91.42**	93.95
MT-ML	**91.21**	**94.41**	91.16	93.95

between the models with the same task. To explain more clearly what models are connected through parameter sharing we illustrate our settings in Figure 1.

(a) SL-ST

	en	es	cs	de
POS	1	2	3	4
NER	5	6	7	8

(b) MT

	en	es	cs	de
POS	1	2	3	4
NER				

(c) ML

	en	es	cs	de
POS	1			
NER	2			

(d) ML-MT

	en	es	cs	de
POS	1			
NER				

Fig. 1. Illustration of how different training modes use all the datasets. E.g. we can see that in MT we have 4 model pairs that share parameters.

Hyperparameters. We used RMSProp [22] optimization algorithm with learning rate 1e−4 with gradient clipping set to 1. Dropout was used before and after LSTM layer and it was set to 50%. For each run we had 60 epochs, each with 512 training steps. Batch size was 32. LSTM hidden and hidden cell size was 300.

Results from this experiments are presented in Tables 2 for NER and Table 3 for POS. We use tag F1 score for NER and tag accuracy for POS. Combination of multilingual and multitask learning managed to beat other learning modes in 4 out of 8 cases. The most significant was 4.1% improvement for Spanish

NER. In all but two cases it beat the single task single language baseline. The result was slight decline of 0.11% was measured for Czech POS. When reviewing these results we noted that there seems to be a negative correlation between the amount of training samples for the task and the improvement we achieved with MT-ML training. This is depicted in Figure 2. The two datasets with highest and lowest number of samples are in fact those with the lowest and highest improvement in score. This indicates that our method is well suited for low resource scenarios but it loses its effectiveness when we have enough training data.

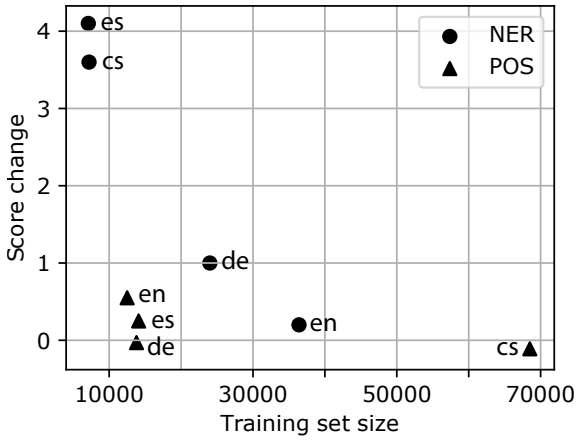

Fig. 2. Relation between training set size and the change in score when using MT-ML instead of ST-SL for each dataset. The score is F1 for NER and accuracy for POS.

4.3 Sharing the output layer

In previous experiments when performing multilingual learning (both ML and MT-ML) the output layers with CRF were not shared across languages. Each language had its own private subset of parameters. Our goal was to let the model learn specifics of each language this way. To confirm our hypothesis that it is beneficial to have private output layers we run the same experiments as before for these two learning modes but this time with only one set of output parameters shared across all four languages. We compare the absolute change in score (F1 for NER, accuracy for POS) in Table 4.

We can see slight improvement in NER (on average +0.17) and slight fall in POS (on average -0.02). The way parts of speech are used in various languages differ more than the way named entities behave. Instinctively this difference in results makes sense. During analysis we noticed that with output layers shared it took longer for the model to converge to near optimal solution in all cases. We think that private output layers make the work easier fort he rest of the model

as they are able to correct model's mistakes. When the output layer is shared the LSTM is forced to predict correct tags as there is no fallback mechanism. It is ultimately able to overcome this challenge but it takes longer because the task solved is harder.

Table 4. The absolute change in score when output layer parameters are shared between languages.

	en	es	de	cs
ML NER	+0.2	+0.8	-0.3	-0.6
MT-ML NER	+0.5	+0.2	+0.2	+0.3
ML POS	-0.06	+0.06	-0.02	-0.04
MT-ML POS	+0.16	-0.06	+0.05	-0.25

5 Future Work and Conclusion

The most important future work is the experimenting with additional languages and also tasks, such as dependency parsing, language modeling and machine translation. We also plan to shift from the multitask learning paradigm to the transfer learning paradigm. Instead of training the model for all available tasks at once, we are interested if we could specialize it only for one specifically selected task (perhaps an extremely low-resource one). To do this we will need an agent capable of dynamically changing the selection policy. Using partially private models [13], adversarial learning [10] or sub-word level representations [23] are several other ideas we plan to experiment with.

So far our model proved itself to be capable of multitask multilingual learning. We were able to beat reasonable single task baselines that use less auxiliary data, especially in small training set cases. Combining data from various heterogeneous sources might be crucial for developing effective solutions especially for low resource languages. We perceive this work as one step towards this goal.

The proposed model work on sentence level which is compatible with many NLP tasks community is solving so far. It can be easily extended by modifying the output layer to solve other tasks. By aggregation we could even combine representations from several sentences to form representations for paragraphs or documents. From these we could gather additional signal for learning as some tasks are traditionally solved in document level fashion, e.g. document classification.

References

1. Agić, Ž., Johannsen, A., Plank, B., Martínez Alonso, H., Schluter, N., Søgaard, A.: Multilingual projection for parsing truly low-resource languages. Transactions of the Association of Computational Linguistics **4**, 301–312 (2016)

2. Ammar, W., Mulcaire, G., Ballesteros, M., Dyer, C., Smith, N.: Many languages, one parser. Transactions of the Association of Computational Linguistics **4**, 431–444 (2016)
3. Benikova, D., Biemann, C., Reznicek, M.: Nosta-d named entity annotation for german: Guidelines and dataset. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014. pp. 2524–2531 (2014)
4. Bos, J., Basile, V., Evang, K., Venhuizen, N., Bjerva, J.: The groningen meaning bank. In: Ide, N., Pustejovsky, J. (eds.) Handbook of Linguistic Annotation, vol. 2, pp. 463–496. Springer (2017)
5. Buys, J., Botha, J.A.: Cross-lingual morphological tagging for low-resource languages. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1954–1964. Association for Computational Linguistics (2016)
6. Conneau, A., Lample, G., Ranzato, M., Denoyer, L., Jégou, H.: Word translation without parallel data. In: 6th International Conference on Learning Representations. Vancouver, Canada (May 2018)
7. Cotterell, R., Heigold, G.: Cross-lingual character-level neural morphological tagging. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 759–770. Association for Computational Linguistics (2017)
8. Gallay, L., Simko, M.: Utilizing vector models for automatic text lemmatization. In: SOFSEM 2016: Theory and Practice of Computer Science - 42nd International Conference on Current Trends in Theory and Practice of Computer Science, Harrachov, Czech Republic, January 23-28, 2016, Proceedings. pp. 532–543 (2016)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)
10. Joty, S., Nakov, P., Màrquez, L., Jaradat, I.: Cross-language learning with adversarial neural networks. In: Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017). pp. 226–237. Association for Computational Linguistics (2017)
11. Kravalova, J., Zabokrtsky, Z.: Czech named entity corpus and svm-based recognizer. In: Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009). pp. 194–201. Association for Computational Linguistics (2009)
12. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 260–270. Association for Computational Linguistics (2016)
13. Liu, P., Qiu, X., Huang, X.: Adversarial multi-task learning for text classification. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1–10. Association for Computational Linguistics (2017)
14. Mikolov, T., Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA. pp. 746–751 (2013)
15. Nivre, J., de Marneffe, M., Ginter, F., Goldberg, Y., Hajic, J., Manning, C.D., McDonald, R.T., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., Zeman, D.:

Universal dependencies v1: A multilingual treebank collection. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016. (2016)
16. Peng, N., Dredze, M.: Multi-task domain adaptation for sequence tagging. In: Proceedings of the 2nd Workshop on Representation Learning for NLP. pp. 91–100. Association for Computational Linguistics (2017)
17. Ruder, S.: A survey of cross-lingual embedding models. CoRR **abs/1706.04902** (2017)
18. Sang, E.F.T.K., Meulder, F.D.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. In: Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003. pp. 142–147 (2003)
19. Søgaard, A., Goldberg, Y.: Deep multi-task learning with low level tasks supervised at lower layers. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 231–235. Association for Computational Linguistics (2016)
20. Subramanian, S., Trischler, A., Bengio, Y., Pal, C.J.: Learning general purpose distributed sentence representations via large scale multi-task learning. In: 6th International Conference on Learning Representations. Vancouver, Canada (May 2018)
21. Tiedemann, J.: Rediscovering annotation projection for cross-lingual parser induction. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. pp. 1854–1864. Dublin City University and Association for Computational Linguistics (2014)
22. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning **4**(2), 26–31 (2012)
23. Yang, Z., Salakhutdinov, R., Cohen, W.W.: Transfer learning for sequence tagging with hierarchical recurrent networks. In: 5th International Conference on Learning Representations. Toulon, France (April 2017)
24. Zirikly, A., Hagiwara, M.: Cross-lingual transfer of named entity recognizers without parallel corpora. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). pp. 390–396. Association for Computational Linguistics (2015)