Using Neural Transfer Learning for Morpho-syntactic Tagging of South-Slavic Languages Tweets

Sara Meftah[*], **Nasredine Semmar**[*], **Fatiha Sadat**[+], **Stephan Raaijmakers**[∓]
[*]CEA, LIST, LVIC, F-91191, Gif-sur-Yvette, France
{sara.meftah, nasredine.semmar}@cea.fr
[+]UQÀM, Montréal, Canada
sadat.fatiha@uqam.ca
[∓]TNO, The Hague, The Netherlands
stephan.raaijmakers@tno.nl

Abstract

In this paper, we describe a morpho-syntactic tagger of tweets, an important component of the CEA List DeepLIMA tool which is a multilingual text analysis platform based on deep learning. This tagger is built for the Morpho-syntactic Tagging of Tweets (MTT) Shared task of the 2018 VarDial Evaluation Campaign. The MTT task focuses on morpho-syntactic annotation of non-canonical Twitter varieties of three South-Slavic languages: Slovene, Croatian and Serbian. We propose to use a neural network model trained in an end-to-end manner for the three languages without any need for task or domain specific features engineering. The proposed approach combines both character and word level representations. Considering the lack of annotated data in the social media domain for South-Slavic languages, we have also implemented a cross-domain Transfer Learning (TL) approach to exploit any available related out-of-domain annotated data.

1 Introduction

Part-of-Speech (POS) tagging is one of the basic and indispensable tasks in any Natural Language Processing (NLP) pipeline; it consists of assigning adequate and unique grammatical categories (Part-of-Speech tags) to words in the sentence. When POS tags are enriched by Morpho-Syntactic Descriptions (MSDs), such as gender, case, tenses, etc. the task is called Morpho-Syntactic Tagging (MST) (Agić et al., 2013). As an example, we provide a Slovene sentence with its MS tags in Figure 1.

MST is a challenging task especially for languages with rich word inflections and free word order like South-Slavic languages. In addition, MST of informal text like social media content of these languages is a more complex task, especially conversational texts. This is due to the conversational nature of the text, the lack of conventional orthography, the noise, linguistic errors, spelling inconsistencies, informal abbreviations and the idiosyncratic style. Also, social media platforms such as Twitter pose an additional issue by imposing 280 characters limit for each tweet.

While recent approaches based on end-to-end Deep Neural Networks (DNNs) have shown promising results for sequence tagging in many languages such as English, much less work has been done on neural models for MST of Slavic languages. In this paper, we evaluate the effect of using neural networks techniques for MST of South-Slavic tweets, where we are faced with a large number of possible word-class tags and only a small hand-tagged in-domain dataset.

NLP neural models with high performance often require huge volumes of annotated data to produce powerful models and prevent over-fitting. Consequently, in the case of social media content, it is difficult to achieve the performances of state-of-the-art models based on hand-crafted features by applying neural models trained on small amounts of annotated data. For this reason, Transfer Learning (TL) was proposed to exploit annotated out-of-domain data-sets. TL aims at performing a task on a target dataset using features learned from a source dataset (Pan and Yang, 2010).

The method presented in this work aims to overcome the problem of the lack of annotated data by significantly limiting the necessary data and instead extrapolating the relevant knowledge from another, related domain. This contribution generalizes previous results for POS tagging of user generated content

Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 235–243
Santa Fe, New Mexico, USA, August 20, 2018.

in social media for five languages: English, French, Italian, German and Spanish (Meftah et al., 2018), by applying our approach on three Twitter corpora of South-Slavic languages: Slovene, Croatian, and Serbian.

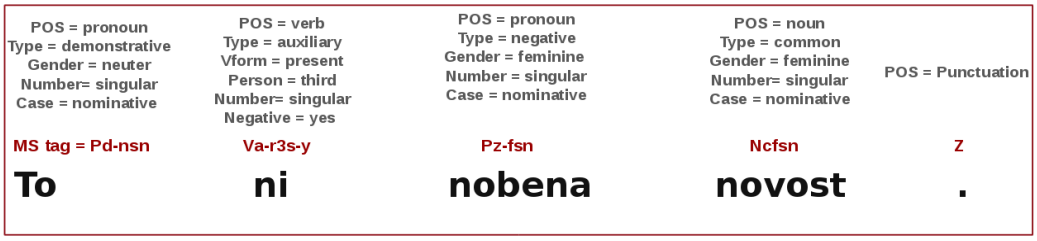

Figure 1: Example of a morphologically-tagged sentence in Slovene: *To ni nobena novost* ("This is not a novelty" in English) .

2 The Model

In this section, we introduce the model we experimented for MS tagging of South-Slavic languages. The model takes as input a tweet T, separated into a succession of n tokens w_i, such as $T = \{w_1, w_2, ..., w_n\}$. The objective is to predict the morpho-syntactic tag $\hat{y}_i$ for each token w_i of the tweet.

2.1 System Architecture

We use a similar architecture to that used in (Meftah et al., 2018) for English, French, Spanish, Italian and German Social Media content's POS tagging, we propose to use a bi-GRU (bidirectional Gated Recurrent Unit) sequence labelling model, preceded by a hybrid word representation. The model architecture is the same among all languages and tasks (Figure 3).

2.1.1 Words Representation

The model learns word-level we_i and character-level ce_i representations respectively for each token x_i, and combines them to get the final representation x_i.

Character level embedding: To capture morphological features, instead of Convolutional Neural Networks (CNNs) used in our previous work (Meftah et al., 2018), we apply in this work a bi-GRU encoder on all characters of each token to induce fully context sensitive character level embedding.

Figure 2 shows the character-level embedding model, a word w_i is divided into a succession of l characters c_i, each defined as a one-hot vector, with value 1 at index c_i and 0 in all other dimensionality, such as w_i will be represented with a $v \times l$ dimensional matrix. Next it's embedded into a $d \times l$ dimensional matrix, where v is the character's vocabulary size of the training set, l is the maximal length of words and d is the character embedding's dimension. Next, a forward GRUs model reads the character vectors from left to right and a backward GRUs model reads characters from right to left. The combination between the last hidden state of the forward GRUs and the last hidden state of the backward GRUs represents ce_i: the character level embedding for the word w_i.

Word-level embedding: we initialize words vectors we_i with FastText (Bojanowski et al., 2016) word embeddings to accurately capture words' semantics.

The combination between character-level embedding and word-level embedding x_i is fed into the bi-GRUs layer.

2.1.2 Bidirectional Gated Recurrent Units

Word vectors $\{x_1, x_2, ..., x_n\}$, which are constructed as a combination of word-level embeddings and character-based representations, are given as input to a 100-dimension bi-GRUs layer which iteratively passes through the sentence in both directions. Let $\overrightarrow{h_t}$ be the GRUs hidden state at time-step t. Formally, a forward GRUs model's unit at a time-step t takes x_t and the previous hidden state $\overrightarrow{h_{t-1}}$ as input, and outputs the current hidden state $\overrightarrow{h_t}$. Each GRUs apply the following transformations:

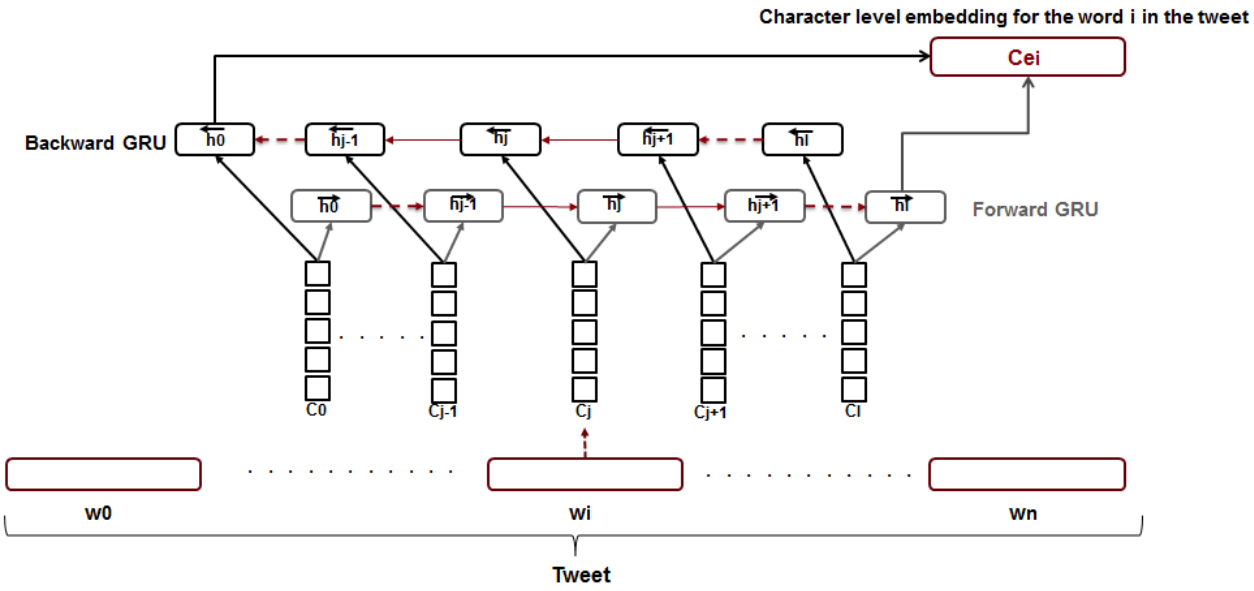

Figure 2: Bi-GRUs layer for character-level embedding.

$$\overrightarrow{r_t} = \sigma(W_{rx}x_t + W_{rh}\overrightarrow{h_{t-1}}) \tag{1}$$

$$\overrightarrow{z_t} = \sigma(W_{zx}x_t + W_{zh}\overrightarrow{h_{t-1}}) \tag{2}$$

$$\overrightarrow{\hat{h}_t} = \tanh(W_{hx}x_t + W_{hh}(\overrightarrow{r_t} \otimes \overrightarrow{h_{t-1}})) \tag{3}$$

$$\overrightarrow{h_t} = \overrightarrow{z_t} \otimes \overrightarrow{h_{t-1}} + (1 - \overrightarrow{z_t}) \otimes \overrightarrow{\hat{h}_t} \tag{4}$$

Here, W's are model parameters of each unit, $\hat{h}_t$ is a candidate hidden state that is used to compute h_t, σ is an element-wise sigmoid logistic function defined as $\sigma(x) = 1/(1 + e^{-x})$, and $\otimes$ denotes element-wise multiplication of two vectors. The update gate z_t controls how much the unit updates its hidden state, and the reset gate r_t determines how much information from the previous hidden state needs to be reset.

In order to take into account the context on both sides of that word, hidden representations $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ from forward and backward units, respectively, are concatenated at every token position, resulting h_t vectors.

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{5}$$

2.1.3 Sequence Labelling

Hidden representations at each time-step are fed through a 80 dimension Fully Connected Layer (FCL) with a ReLU activation, followed by a final dense layer with a softmax activation to generate a probability distribution over the output classes at each time-step.

3 Neural Transfer Learning Methodology

Neural transfer learning is applied to address the problem of the need for annotated data for morpho-syntactic tagging of social media texts. It consists of learning a parent neural network on a source problem with enough data, then transferring a part of its weights to represent data of a target problem with few training examples.

In this work, we experiment with cross-domain transfer; knowledge is transferred from a source domain to a target domain. In our case, the source domain is a standard text corpus of a language and the

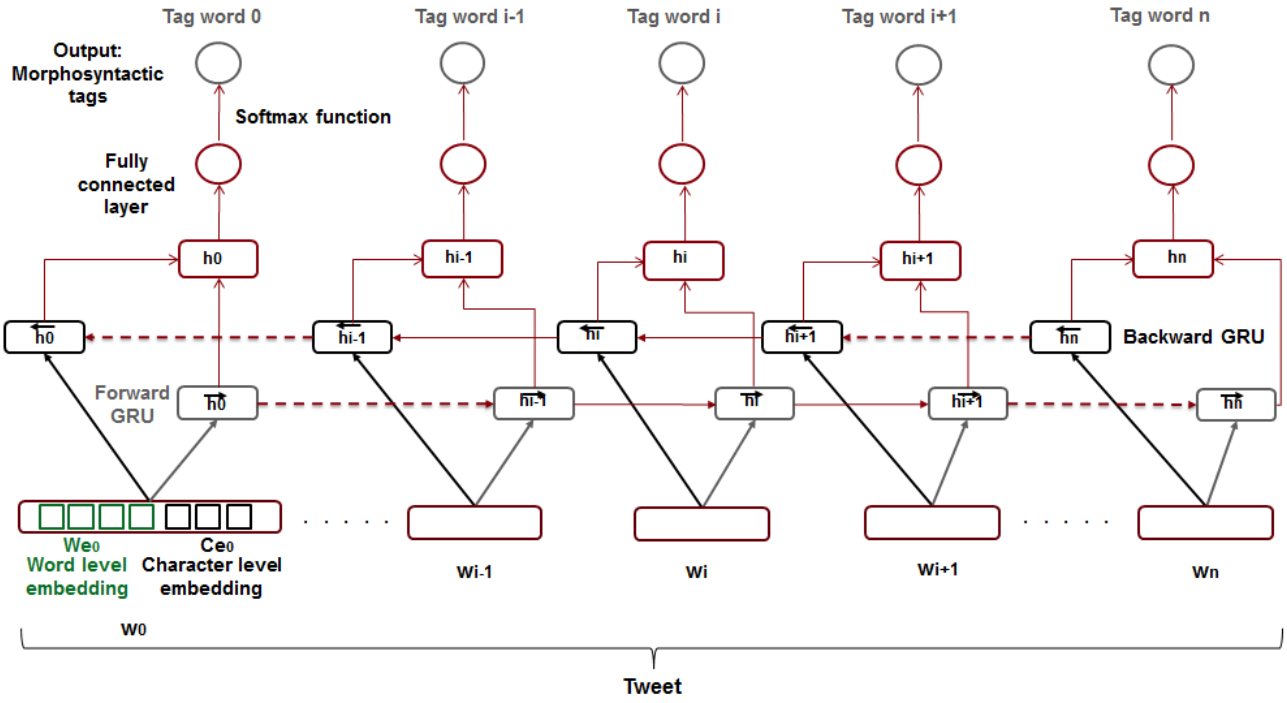

Figure 3: Overall system architecture. First, the system embeds each word of the current sentence into two representations: a character-level representation using bi-GRUs and a word-level representation. Then, the two representations are combined and fed into a bi-GRUs layer, the resulting vector is fed to a fully connected layer and finally a softmax layer to perform MS tagging.

target domain is the Twitter text of the same language. The source and the target problems are trained for the same task (MST), even if source and target data-sets do not share the same tag-set.

As illustrated in Figure 4, we have a parent neural network N_p with a set of parameters θ_p split into two sets: $\theta_p = (\theta_p^1, \theta_p^2)$, and a child network N_c with a set of parameters θ_c split into two sets: $\theta_c = (\theta_c^1, \theta_c^2)$.

(1) We learn the parent network on annotated data from the source problem on a source dataset D_s.

(2) We transfer weights of the first set of parameters of the parent network N_p to the child network N_c: $\theta_c^1 = \theta_p^1$.

(3) Then, the child network is fine-tuned to the target problem by training it on the target data-set D_c.

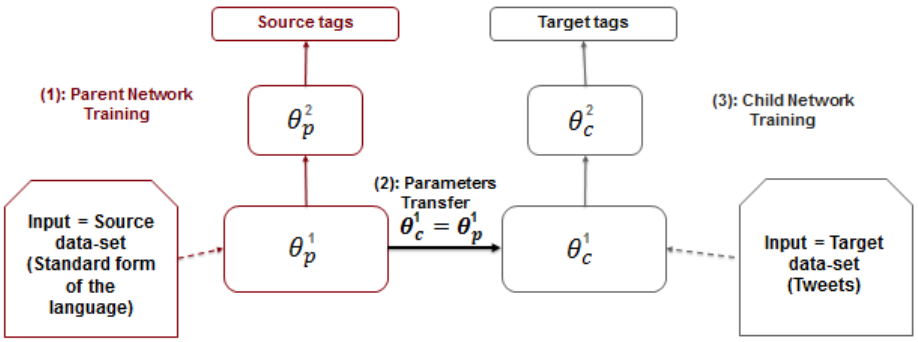

Figure 4: Cross-domain transfer learning scheme for morpho-syntactic tagging.

Language	Domain	Corpus	Tag-set size	# Sentences	# Tokens
Slovene	Out of domain	All	1,304	27,829	586,248
	In domain	Train	758	3,934	37,756
		Dev	422	713	7,056
		Test	598	2023	19,296
Croatian	Out of domain	All	772	24,611	506,460
	In domain	Train	580	4,089	45,609
		Dev	363	791	8,886
		Test	487	1,883	21412
Serbian	Out of domain	All	557	3891	86765
	In domain	Train	575	3463	45,708
		Dev	385	737	9581
		Test	498	1,684	23327

Table 1: Statistics of the different source and target data-sets.

4 Experiments Setup

4.1 Task and Data Description

Two types of data-sets were provided in the MTT shared task (Zampieri et al., 2018) for each language: (1) a small manually annotated Twitter data-set (in-domain data) (Erjavec et al., 2017; Ljubešić et al., 2017a; Ljubešić et al., 2017b); (2) a large manually annotated raw canonical data-set (out-of-domain data) (Erjavec et al., 2015; Ljubešić and Klubička, 2014)[1].

The statistics of the data-sets are described in table 1. All corpora are in the CoNLL format. They are already tokenized. Each token in a tweet is associated with a single morpho-syntactic tag using different alphabetical characters for denoting different category values. The first letter represents POS tag, while other tag positions represent morpho-syntactic categories like case, genre, etc. For instance, the MS tag *Ncfsn* of the word *novost* in the example, provided in figure 1, would denote a noun, common, feminine, singular, nominative token.

4.2 Transfer Learning Experiments

Cross-domain TL is evaluated on the three languages: Slovene, Serbian and Croatian, following three main phases: (1) training the parent network on the source problem on rich out-of-domain data, (2) transferring weights of the first set of parameters to the target problem (these weights are used to initialize the child model's first set of parameters, rather than starting from a random position[2]), and finally (3) fine-tuning the child network on low-resource in-domain data.
Our experiments have shown that using a smaller learning rate for weights that will be fine-tuned (first set of weights), in comparison to the randomly initialized weights (second set of weights) leads to slightly improvements.

4.3 Implementation Details

All experiments described in this section are implemented using the PyTorch deep learning library. We use the Stochastic Gradient Descent (SGD) optimizer with momentum of Nesterov (Sutskever et al., 2013) in all experiments. We set the character embedding dimension at 50, the dimension of hidden states of the character level embeddings GRUs layer at 80, 100 for sequence labelling GRUs layer and FCL dimension at 80. We use dropout training with probability 0.3 before the input to GRUs and FCL layers in order to avoid overfitting.

Tokens are lowercased while the character-level component still retains access to the capitalization information. Word embeddings were set to size 300, pre-loaded from publicly available FastText pre-trained vectors on common crawl[3]. Word level embeddings are fine-tuned during training. The training was performed in batches of 64 sentences for parent models training and 32 sentences for child networks

[1] A large automatically annotated web data was also provided by the shared task organizers, but we did not make use of it in this work.

[2] The weights of the second set of parameters of the child model are randomly initialized.

[3] https://github.com/facebookresearch/fastText/blob/master/docs/crawl-vectors.md

training, and was stopped if development set accuracy did not improve for 4 epochs. The best overall model on the development set was then used to report performance on the test data.

5 Results

In this section, we report the results of our system described in section 3. Firstly, we report the results of our system submitted to the MTT shared task. Thereafter, we present some improvements done on our model after the submission of the results for this task.

Language	Slovene	Croatian	Serbian
Acc. without transfer learning (%)	79.8	80.3	81.2
Acc. with transfer learning - A (%)	82.6	82.9	82.1
Acc. with transfer learning - B (%)	**83.36**	**83.87**	**82.54**

Table 2: Our system accuracy (acc.) with and without cross-domain transfer learning (A represents the results of the system submitted to the shared task, and B an improved performance after the submission).

In Table 2, we compare the performances of the neural network model described in section 3 trained only on target data-set (first line) against the neural network trained with TL (second line). We can see that the TL method significantly improves results on all languages. Table 2 further shows that the improvements made by cross-domain TL for Slovene and Croatian (+2.8% , +2.7%) are more important than improvements made by cross-domain TL for Serbian (+0.9%). This phenomenon can be explained by the fact that as illustrated in Table 1, the source data-set for Serbian experiments is very small compared to source data-sets for Slovene and Croatian, hence the improvement is less substantial.

In the above experiments submitted to the MTT shared task, we transfer the parameters of the parent network when they achieve the highest performance on development set of the out-of-domain data-set. However, as shown in previous studies on TL (Mou et al., 2016), the parameters perfectly trained on a source data-set may be too specific to it, hence, the model may underfit on the target data-set. We therefore made more experiments to pick the parent model trained on the ideal epoch for the target data-sets for further fine-tuning. In the third line in table 2, we give the highest performance of TL on target data-sets.

6 Discussion

6.1 Ablation Study

In order to assess the importance of embeddings to handle the problem of Out Of Vocabulary words (OOVs), we have conducted a series of experiments through ablating one layer each time, character-level embedding and word-level embedding and observing how that affects the performance.

For these experiments, we use the neural model described in section 2 without TL technique (i.e., only in-domain data).

In table 3, we report the results of our experiments on Slovene data-set. In each column, we provide the results on a different set of tokens. In the first, we used all tokens of the test set, in the second, only In Training Vocabulary words (ITVs), i.e words that have been seen in the training set, in the third, Out Of Training vocabulary words (OOTVs), in the fourth, In Embedding Vocabulary words (IEVs), i.e words that have been found in the FastText pre-trained words vectors, and in the fifth, Out Of Embedding Vocabulary words (OOEVs).

We provide for each set of tokens, the number of tokens in the first line, the vocabulary size in the second, the simple neural model's accuracy in the third, the accuracy of the neural network without character-level embedding in the fourth and the accuracy of the neural network without pretrained word-level embedding (i.e, words embeddings are randomly initialized) in the fifth.

We notice in table 3 a significant gap between the overall accuracy on ITVs (89.78%) and the accuracy on OOTVs (45.92%). We can also see that removing character-level embedding drops significantly the overall accuracy (-7%). However, the accuracy on OOTVs drops by 27% (33.40% error reduction) while the one for ITVs drops only by 1.5% (13.7% error reduction), which confirms the effectiveness of character-level embedding on OOTVs.

	All tokens	ITVs	OOTVs	IEVs	OOEVs
# tokens	19,296	14,828	4.468	17,045	2,251
Vocabulary size	6,538	2,400	4,138	4,850	1,688
Neural model (NM) acc. (%)	79.8	89.78	45.92	80.86	70.27
NM without character embedding acc. (%)	72.16	88.23	18.80	74.23	56.46
NM without pre-trained words embedding acc. (%)	76.42	88.67	36.05	77.88	65.97

Table 3: Ablation study on character-level embedding and pre-trained words embedding for Slovene tweets MS tagging.

6.2 Impact of Transfer Learning

In the experiments below, we are interested on the impact of using TL on Slovene data-set. In table 4, we compare the performance of our model on the full MSD features and the performance on only POS tags. The first two columns give token-level and the sentence-level accuracy without using TL, and the second two columns give token-level and the sentence-level accuracy using TL. The first line shows the accuracies on all MSD features (the overall accuracy), the second one gives the accuracies on only POS tags.

Table 4 shows that POS accuracy is quite high compared to the full accuracy, this is due to the small POS tag-set (13 POS tags). In addition to that, POS ambiguity of Slovene words is relatively low conversely to the other MSD features. We can observe an improvement of 4.56% brought by TL on the token-level full accuracy, and 2.16% on POS tags. This means that 50% of the error reduction made by TL was on POS tags.

	Without transfer learning		With transfer learning	
	Token acc. (%)	**Sentence acc. (%)**	**Token acc. (%)**	**Sentence acc. (%)**
All MSD features	79.8	27.92	83.36	33.61
Only POS Tags	89.55	45.72	91.71	53.97

Table 4: Comparison between our model accuracy on Slovene on the full MSD features and on POS tags.

Table 5 gives the results for the Slovene data-set first tagged without TL and then using TL. The first two columns give the numbers and the percentage of tokens that have their POS tags changed by the model using TL compared to the model without TL, and the second two columns give the numbers and the percentage of tokens with changed morpho-syntactic tags (including POS tags). The first line shows the tags that were wrong, but the TL changed to the correct ones, the second gives the numbers of those tokens which the standard neural network tagged correctly, but the TL technique falsified. The third line shows those instances where the tag was not changed by the TL technique. The last line shows the number of tokens that have tag assigned that was subsequently changed by the TL technique into a wrong tag.

	POS tags		Morphosyntactic descriptions	
	# Token	**Percentage (%)**	**# Token**	**Percentage (%)**
Corrected	949	4.91	1,402	7.26
Falsified	532	2.75	682	3.53
Identical	17,459	9047	15,789	81.82
Confused	356	1.84	1,423	7.37

Table 5: Modifications made by transfer learning on Slovene data-set.

We can observe that 7.26% of MSD tags were corrected by the TL technique and 3.53% were falsified. In table 6, we investigate which POS tags have benefited the most from the cross-domain TL technique. The first column presents the number of tokens of each POS tag on the test-set, the second (the third) set of columns gives the accuracy, number of true positives (TP), number of false negatives (FN) and false positives without using TL (with TL). We can observe a significant accuracy improvement on adjectives (+10%), nouns (+6%) and adverbs (+2%), with a drop of accuracy on abbreviations and interjections (-8%).

		Without transfer learning				With transfer learning			
POS tags	**# tokens**	Accuracy (%)	# TP	# FN	# FP	Accuracy (%)	# TP	# FN	# FP
X (Residual)	1782	88.10	1570	212	150	90.62^{+}	1615	167	217
Q (Particle)	1011	89.02	900	111	78	89.61^{+}	906	105	88
R (Adverb)	1548	85.01	1316	232	265	87.33^{+}	1352	196	211
Z (Punctuation)	2872	99.68	2863	9	6	99.75^{+}	2865	7	5
N (Noun)	2808	79.98	2246	562	551	86.36^{+}	2425	383	379
V (Verb)	3427	92.64	3175	252	357	94.28^{+}	3231	196	245
P (Pronoun)	1652	88.68	1465	187	149	90.25^{+}	1491	161	129
C (Conjunction)	1571	96.37	1514	57	59	96.05^{-}	1509	62	49
A (Adjective)	852	69.71	594	258	281	79.34^{+}	676	176	151
S (Adposition)	1096	96.98	1063	33	33	97.71^{+}	1071	25	25
Y (Abbreviation)	107	87.85	94	13	23	86.91^{-}	93	14	30
M (Numeral)	334	89.82	300	34	22	90.41^{+}	302	32	34
I (Interjection)	236	76.69	181	55	41	68.64^{-}	162	74	35

Table 6: The impact of transfer learning on POS tags prediction on Slovene data-set.

6.3 Improvements when Using Conditional Random Fields

In this section, we report new improvements of the performance of our model after the submission to the MTT shared task. Instead of using a softmax function in the topmost of the model for inference (MS tagging), as described in the section 2, we use Conditional Random Fields (CRFs) (Lafferty et al., 2001) layer to decode the best tag sequence from all possible tag sequences with consideration of outputs from bi-GRUs layer and correlations between surroundings labels.

Indeed, it has been shown that CRFs are more appropriate for sequence labelling tasks and can produce higher performances (Huang et al., 2015; Ma and Hovy, 2016). Table 7 shows a significant improvement on the performances of our system for all languages, by replacing the softmax function by a CRFs layer (Ma and Hovy, 2016).

	Slovene	**Croatian**	**Serbian**
Model + softmax acc. (%)	83.36	83.87	82.54
Model + CRFs acc. (%)	**86.23**	**87.47**	**86.4**

Table 7: Comparison between the model's performances using Softmax and CRFs layers.

7 Conclusion

In this paper, we have presented a neural network model using Transfer Learning (TL) for Morpho-syntactic (MS) tagging of Twitter texts. In particular, we have conducted experiments on tweets for three South-Slavic languages: Slovene, Croatian and Serbian. We have more specifically used an approach which combines both character and word level representations. The obtained results show that TL improves the performance of the MS tagging task for the three involved languages.

This work leaves two important open issues, which certainly deserve further research. First, we intend to apply our model to other morphologically rich languages like MSA and its dialects. The second perspective consists in modelling the lexical and syntactic similarities between the source and target languages (domains) in order to incorporate this external linguistic knowledge in the neural network model.

8 Acknowledgments

This research work is supported by the ASGARD project. This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 700381.

References

Željko Agić, Nikola Ljubešić, and Danijela Merkler. 2013. Lemmatization and morphosyntactic tagging of croatian and serbian. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, pages 48–57.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Tomaž Erjavec, Nikola Ljubešić, and Nataša Logar. 2015. The slWaC Corpus of the Slovene Web. *Informatica*, 39(1):35–42.

Tomaž Erjavec, Darja Fišer, Jaka Čibej, Špela Arhar Holdt, Nikola Ljubešić, and Katja Zupan. 2017. CMC training corpus janes-tag 2.0. Slovenian language resource repository CLARIN.SI.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Nikola Ljubešić and Filip Klubička. 2014. {bs,hr,sr}wac - web corpora of bosnian, croatian and serbian. In *Proceedings of the 9th Web as Corpus Workshop (WaC-9)*, pages 29–35. Association for Computational Linguistics.

Nikola Ljubešić, Tomaž Erjavec, Maja Miličević, and Tanja Samardžić. 2017a. Croatian twitter training corpus ReLDI-NormTagNER-hr 2.0. Slovenian language resource repository CLARIN.SI.

Nikola Ljubešić, Tomaž Erjavec, Maja Miličević, and Tanja Samardžić. 2017b. Serbian twitter training corpus ReLDI-NormTagNER-sr 2.0. Slovenian language resource repository CLARIN.SI.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Sara Meftah, Nasredine Semmar, and F Sadat. 2018. A neural network model for part-of-speech tagging of social media texts. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.*

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in nlp applications? *arXiv preprint arXiv:1603.06111*.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. 2013. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. Language Identification and Morphosyntactic Tagging: The Second VarDial Evaluation Campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Santa Fe, USA.