Punctuation Restoration System for Slovene Language

Marko Bajec[1](✉), Marko Janković[1,2], Slavko Žitnik[1], and Iztok Lebar Bajec[1]

[1] Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia
Marko.bajec@fri.uni-lj.si
[2] Vitasis d.o.o., Rakek, Slovenia

Abstract. Punctuation restoration is the process of adding punctuation symbols to raw text. It is typically used as a post-processing task of Automatic Speech Recognition (ASR) systems. In this paper we present an approach for punctuation restoration for texts in Slovene language. The system is trained using bi-directional Recurrent Neural Networks fed by word embeddings only. The evaluation results show our approach is capable of restoring punctuations with a high *recall* and *precision*. The F1 score is specifically high for *commas* and *periods*, which are considered most important punctuation symbols for the understanding of the ASR based transcripts.

Keywords: Punctuation restoration · Automatic speech recognition · Text processing

1 Introduction

The goal of punctuation restoration is to identify positions in raw text where punctuation symbols are missing or could be added to improve the readability and semantic value of text. It is typically used in combination with ASR systems that produce sequences of words without any punctuation symbols. The text is then improved by the insertion of punctuation symbols to positions where they fit according to specific rules.

The ability to accurately restor punctuations is found to be very important, not just for an understanding of the recognized text, but also for further processing such as providing quality translations. In this paper we describe an approach for punctuation restoration that uses bi-directional neural networks for punctuation predictions.

The rest of the paper is structured as follows: Sect. 2 provides related works, Sect. 3 the method used, Sect. 4 the dataset that we used and Sect. 5 the evaluation results. Short conclusion is given in Sect. 6.

2 Related Works

The challenge of punctuation restoration is not new. A lot of efforts have been made to restore punctuation symbols automatically. In terms of how the punctuation restoration

F. Dalpiaz et al. (Eds.): RCIS 2020, LNBIP 385, pp. 509–514, 2020.
https://doi.org/10.1007/978-3-030-50316-1_31

problem is modeled, the existing approaches can be categorized into three categories [1]. The first category comprises approaches that model punctuations as hidden inter-word states and use *n-gram* language models or *hidden Markov chains* to restore punctuations [2]. Approaches in the second category deal with the punctuation restoration as with a *sequence labeling task* where labels are punctuations, and words are objects to which labels are assigned. It has been shown in the literature that feature-rich models such as *Condition Random Fields* (CRFs) are specifically suited for the task at hand. Approaches that are based on CRFs achieve F1 scores around 55% for English datasets [3]. In the third category there are approaches that use *neural networks* (NN), specifically *deep neural networks* (DNN) to predict punctuations in text.

As for many other Natural Language Processing (NLP) tasks, it has been shown also for the punctuation restoration problem that NNs outperform other known approaches. Several NN architectures have been proposed in the literature for this purpose. Improvements in punctuation restoration accuracy have been demonstrated with different NN-based architectures, such as *convolutional* NN [4], *long short-term memory* NN [5], *bi-directional recurrent* NN with *attention mechanism* [6], *recurrent* NN *encoder-decoder* architecture with *attention layer* [7], etc.

In this paper we focus on the punctuation restoration for Slovene language. The approach that we describe is based on NN, with a bi-directional recurrent architecture and attention mechanism. On the input we only use lexical features, including *word embeddings*. We are not aware of any prior work that would address the problem of punctuation restoration for Slovene texts. The only known attempt that we are aware of was focused on *comma replacement* and *correction* [8] employing various machine learning techniques. As we show in this paper, our approach is superior as it achieves considerably higher accuracy for the comma prediction problem. Moreover, the prediction accuracy of punctuation restoration is in general better for Slovene texts than it is usually reported for English (cf. [3–7]). Although this comparison does not necessarily make sense, as linguistic features differ from language to language, it may represent an interesting observation, worthy of further research. Most importantly, the prediction accuracy is in particular high for commas and periods that are considered the most important punctuation symbols for improving human readability of ASR generated texts and its further machine processing.

3 Method

The model that we use in our approach is built along the NN architecture suggested by O. Tilk and T. Alumäe in [6]. In simple words, the model works as follows: at each step t the model calculates the probabilities of missing punctuation symbols p_t between the current input word x_t and previous input word x_{t-1}. The sequence of words $\boldsymbol{X}=(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_T)$, in which each word is represented as a *one-hot encoded* vector $\boldsymbol{x}_i$, is first processed by two recurrent layers. One processing the sequence in forward and the other in backward direction. Optionally, if we want pre-trained word embeddings to replace one-hot encoded word vectors, the two recurrent layers are preceded by a *shared embedding layer* with weights $\boldsymbol{W}_e$.

The hidden state at step *t* of the forward recurrent layer is calculated with GRU where *tanh* is used for the activation function: $\boldsymbol{hidden\_f}_t = GRU\left(\boldsymbol{hidden\_f}_{t-1}, \boldsymbol{x}_t * \boldsymbol{W}_e\right)$.

The hidden state of the backward recurrent layer $\boldsymbol{hidden\_b}_t$ is computed in the same way, except that for a reverse input sequence order. Then both hidden states $\boldsymbol{hidden\_f}_t$ and $\boldsymbol{hidden\_b}_t$ are concatenated: $\boldsymbol{hidden}_t = concat\left(\boldsymbol{hidden\_f}_t, \boldsymbol{hidden\_b}_t\right)$.

Like suggested in [6], the bi-directional recurrent layer is proceeded by a unidirectional GRU layer with an attention mechanism. While the GRU layer sequentially processes the states of the previous layers and keeps track of the position in the sequence, the attention mechanism focuses on potential relationships among words before and after the current position, signaling important information for the punctuation decisions. The output state $\boldsymbol{s}$ at step t is then calculated as $\boldsymbol{s}_t = GRU(\boldsymbol{hidden}_t, \boldsymbol{s}_{t-1})$. The state $\boldsymbol{s}$ is finally *late fused* [12] into $\boldsymbol{f}_t$ and fed to the output layer.

The punctuation probabilities $\boldsymbol{y}_t$ at position t are calculated using *Softmax* function as follows: $\boldsymbol{y}_t = Softmax\left(\boldsymbol{f}_t * \boldsymbol{W}_y + \boldsymbol{b}_y\right)$, where $\boldsymbol{b}_y$ is a bias vector.

4 Dataset and Data Preparation

To train NN, large datasets are usually required. In our case we used the corpus Gigafida 2.0, which represents a reference corpus of written Slovene [9]. The corpus is comprised of articles from newspapers, magazines, a selection of web texts, and excerpts from different types of publications, i.e. fiction, schoolbooks, and non-fiction. Altogether it includes 60 million sentences out of which 40 million were used for training.

Prior to the training process, we labeled sentences in the corpus with information on *predictive classes*, i.e. which tokens represent punctuation symbols that we would like to learn how to predict. We did that by replacing punctuations with special labels, defined for each punctuation symbol separately. By these transformations we assured that punctuation symbols would be treated as separate tokens rather than as being additional characters of words. In addition, we decapitalized all sentences, since this information is not available in ASR generated texts and we did not want the NN to become dependent on it.

In the data preparation phase, we also trained word embeddings. We did the training from scratch (using GloVe [14]) since all the embeddings that are available for Slovene language capture semantic similarity of words only.

5 Experiments and Results

5.1 Punctuation Restoration Accuracy

For the experimentation, 10% of the Gigafida 2.0 dataset was reserved for validation and testing and the rest for training. We experimented with several word representations, including one-hot encoding, pre-trained GloVe embeddings for Slovene language, and GloVe embeddings, specifically trained for the purpose of punctuation restoration.

For the NN hyperparameters, such as *learning rate* and the *number of hidden layers*, we followed suggestions reported in related works. We used 256 hidden layers and a learning rate of 0.02.

The results are shown in the table below. The numbers noWe column correspond to experiments, in which we did not use word embeddings. The vocabulary was created from 200.000 most frequently used words in the corpus. Columns PreWe and SpecWe with pre-trained and newly trained GloVe embeddings, respectively.

As it can be noticed, best performance can be achieved with embeddings that are specifically trained for the punctuation restoration task. This was expected, as words in the pre-trained embeddings appear in their *lemmatized* form, while for the punctuation restoration we assume it is better to distinguish between different forms of the same word. Slovene is one of the languages where word forms may depend on cases – a feature that is known for Balto-Slavic languages plus few others, like German. In German, cases are mostly marked on articles and adjectives, while in Balto-Slavic languages they are marked on nouns. With lemmatization we lose this information, which might not be irrelevant for predicting punctuation positions in a text (Table 1).

Table 1. Punctuation restoration results

Punctuation	Precision			Recall			F1 measure		
	noWe	PreWe	SpecWe	noWe	PreWe	SpecWe	noWe	PreWe	SpecWe
Comma	0.776	0.864	**0.905**	0.692	0.798	**0.872**	0.731	0.830	**0.888**
Period	0.661	0.798	**0.862**	0.545	0.834	**0.869**	0.598	0.816	**0.865**
Ques. m.	0.352	0.655	**0.722**	0.027	0.487	**0.527**	0.050	0.559	**0.609**
Exc. m.	–	–	**0.520**	–	–	**0.030**	–	–	**0.070**
Overall	*0.731*	*0.823*	***0.881***	*0.620*	*0.799*	***0.859***	*0.671*	*0.811*	***0.870***

5.2 Comparison with the Best-Known Results for Slovene

In Table 2 we compare our results with the best-known results for punctuation restoration in Slovene texts. We are not aware of any other work to compare with except for the one in [8], where the authors deal with the problem of *comma placement* and *correction* while other punctuations are not considered. Hence, the comparison was performed only for punctuation symbol *comma*. In [8] the authors evaluate various machine learning approaches by using *grammar-based features* that they generate specifically for the problem at hand. The classification methods they test are *random forests*, *support vector machines*, *naive Bayesian classifier*, *RBF network*, *alternating decision trees*, *AdaBoost.M1*, and *decision table*. The best performing methods are random forests, alternating decision trees, and decision table. The results in Table 2 show that our model significantly improves accuracy for the *comma placement* problem on the same dataset, i.e. Šolar[1].

5.3 Comparison with Punctuation Restoration Accuracy for Other Languages

Table 3 provides results of punctuation restoration for three different languages, Estonian, English and Slovene. The results for English and Estonian are taken from [6].

[1] http://eng.slovenscina.eu/korpusi/solar.

Table 2. Comparison between supervised ML approaches and bi-directional RNN with attention mechanism for comma restoration problem on the dataset Šolar

Punctuation	Precision	Recall	F1
NaiveBayes	0.269	0.861	0.410
RandomForest	0.913	0.542	0.680
ADTree	0.916	0.426	0.581
DecisionTable	**0.920**	0.577	0.709
Bi-directional RNN	0.890	**0.837**	**0.863**

Table 3. Comparison between languages

Language	Precision		Recall		F1	
	Comma	Period	Comma	Period	Comma	Period
English	0.655	0.733	0.471	0.725	0.548	0.729
Estonian	0.816	0.738	0.754	0.773	0.784	0.755
Slovene	**0.905**	**0.862**	**0.872**	**0.869**	**0.888**	**0.865**

Even though the results do not reveal much about the methods used, as they were in all the three cases similar, i.e. a bi-directional RNN with attention mechanism, it is interesting to notice that for Slovene language we can achieve a much higher punctuation restoration accuracy than for the other two languages. The relatively high difference can be attributed to different reasons, but we believe it indicates that the Slovene grammar is relatively rich comparing to English and Estonian language in terms of information the NN can exploit when learning how to predict punctuation places in Slovenian text.

6 Conclusion

Punctuation restoration is an important process that is typically used in combination with ASR systems to place missing punctuations in the output text of the recognizer. The punctuations that set sentence boundaries are particularly important, as they improve the readability of the output text and facilitate further machine processing, such as for example machine translation.

In this paper we focused on the problem of punctuation restoration for Slovene language that has not been addressed yet in this manner. Taking into account the findings of related works that deal with the same challenge on other languages, we employed DNN to predict missing punctuations. The NN architecture uses recurrent GRU gates in bi-directional mode, plus an attention mechanism to give the network additional context information for punctuation decisions. The evaluation results show the suggested model is able to achieve a much higher prediction accuracy than previously evaluated machine learning techniques. More surprisingly, the results demonstrate using nearly the same

NN architectures significantly higher prediction accuracy can be achieved for Slovene than for English or Estonian language.

References

1. Yi, J., Tao, J.: Self-attention based model for punctuation prediction using word and speech embeddings. In: Proceedings of ICASSP 2019 – 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7270–7274 (2019)
2. Stolcke, A., et al.: Automatic detection of sentence boundaries and disfluencies based on recognized words. In: IC-SLP 1998, Sydney (1998)
3. Ueffing, N., Bisani, M., Vozila, P.: Improved models for automatic punctuation prediction for spoken and written text. In: INTERSPEECH, pp. 3097–3101 (2013)
4. Che, X.et al.: Punctuation prediction for unsegmented transcript based on word vector. In: Proceedings of the LREC, pp. 654–658 (2016)
5. Tilk, O., Alumae, T.: LSTM for punctuation restoration in speech transcripts. In: INTERSPEECH, pp. 683–687 (2015)
6. Tilk, O., Alumae, T.: Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In: INTERSPEECH, pp. 3047–3051 (2016)
7. Klejch, O., Bell, P., Renals, S.: Sequence-to-sequence models for punctuated transcription combining lexical and acoustic features. In: ICASSP, pp. 5700–5704 (2017)
8. Krajnc, A., Robnik-Sikonja, M.: Postavljanje vejic v Slovenščini s pomočjo strojnega učenja in izboljšanega korpusa Šolar. In: Darja Fišer slovenščina na spletu in v novih medijih, pp. 38–43 (2015)
9. Logar, N.: Reference corpora revisited: expansion of the Gigafida corpus. In: Gorjanc, V., et al. (eds.) Dictionary of modern Slovene: problems and solutions (Book series Prevodoslovje in uporabno jezikoslovje), 1st edn. Ljubljana University Press, Ljubljana, pp. 96–119 (2017)
10. Luong, T., Hieu, P., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1412–1421, Lisbon. Association for Computational Linguistics (2015)
11. Yuan, G., Glowacka, D.: Deep gate recurrent neural network. In: Proceedings of ACML, pp. 350–365 (2016)
12. Snoek, C.G., Worring, M., Smeulders, A.W.: Early versus late fusion in semantic video analysis. In: Proceedings of the 13th Annual ACM International Conference on Multimedia, pp. 399–402. ACM (2005)
13. Khattak, F.K., Jeblee, S., Pou-Prom, C., Abdalla, M., Meaney, C., Rudzicz, F.: A survey of word embeddings for clinical text. J. Biomed. Inform.: X **4**, 100057 (2019). ISSN 2590-177X
14. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543, Doha. Association for Computational Linguistics (2014)