CONTEXT BASED LEMMATIZER FOR POLISH LANGUAGE *

Michal Karwatowski
UST-AGH
Cracow
mkarwat@agh.edu.pl

Marcin Pietron
UST-AGH
Cracow
pieton@agh.edu.pl

ABSTRACT

Lemmatization is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word’s lemma, or dictionary form. In computational linguistics, lemmatisation is the algorithmic process of determining the lemma of a word based on its intended meaning. Unlike stemming, lemmatisation depends on correctly identifying the intended part of speech and meaning of a word in a sentence, as well as within the larger context surrounding that sentence. As a result, developing efficient lemmatisation algorithm is the complex task. In recent years it can be observed that deep learning models used for this task outperform other methods including machine learning algorithms. In this paper the polish lemmatizer based on Google T5 model is presented. The training was run with different context lengths. The model achieves the best results for polish language lemmatisation process.

Keywords NLP · Google T5 · Deep learning · lemmatisation · sequence-to-sequence models

1 Introduction

Natural Language Processing consists of many tasks, the role of each is extracting and processing human understandable meaning from the text data. Some tasks like classification encompass the complete flow from data to answer, in other tasks like part of speech tagging, results are often used as an input for next algorithms. An interesting and complex problem is translation, where the meaning of the text needs to be extracted and encoded back to the text in a different language. This approach describes a family of NLP tasks called text-to-text or sequence-to-sequence processing. Another example of text-to-text processing is lemmatisation, it finds a base form of a given word or expression. Complexity of this problem varies from language to language. In English the number of word variations is usually low, there are simple rules and not many exceptions. However in Slavic languages such as Polish inflection of words it is significantly more complicated and effective lemmatisation is beyond capabilities of a rule based or edit tree classification methods [1], [2]. Situation becomes more difficult when we include multi-segment expressions. For example: “ministerstwie kultury i dziedzictwa narodowego” should be lemmatized to “ministerstwo kultury i dziedzictwa narodowego” not “ministerstwo kultura i dziedzictwo narodowy”. In such cases context and inflection of words should be taken for correct lemmatisation. Another issue are foreign phrases changed using Polish grammar, which is especially common with names and places. The next problem in lemmatisation arise from encountering previously unseen words during inference time. In recent years it can be observed that sequence-to-sequence models used for lemmatisation of highly inflective languages outperform other methods [3]. In this paper the Google T5 model was trained on polish language dataset for PolEval competition. The training was run with three configurations: no context, fixed and variable context lengths. The experiments show the efficiency of fine-tuned Google T5 based lemmatiser model on polish language and the impact the context on model accuracy. The context based Google T5 model achieves best results for polish language which were measured so far. The simulations were run with small and base T5 models.

* Citation: Authors. Title. Pages... DOI:000000/11111.
2 Related work

There were a few approaches to Polish language lemmatisation, for example KRNNT tagger [4] based on recurrent neural network. In 2019 lemmatisation was a part of a PolEval competition [4]. The most common machine learning approaches for lemmatisation are based on edit tree classification, where all possible edit trees or word-to-lemma transformation rules are gathered from the training data. Then a classifier is trained to choose the correct one for a given input word. Many recent works build on the sequence-to-sequence learning paradigm [5], [6], [1], [7], [8] and [9]. Bergmanis and Goldwater [5] present the Lematus context-sensitive lemmatisation system, where the model is trained to generate the lemma from a given input word one character at a time. The Lematus system outperforms other context-aware lemmatisation systems. [3] Sequence-to-sequence models have also been applied in the context of morphological reinflection, the reverse of the lemmatisation task. In the CoNLL-SIGMORPHON 2017 Shared Task on Universal Morphological Reinflection [10] several of the top-ranking systems were based on sequence-to-sequence learning [11], [5].

3 Model architecture and training setup

3.1 Text-to-text model

Google T5 [12] is an extremely large new neural network model that is trained on a mixture of unlabeled text and labeled data from popular natural language processing tasks. Google’s T5 is one of the most advanced natural language models to date. It builds on top of previous work on Transformer based models. Unlike BERT, which had only encoder blocks, and GPT-2, which had only decoder blocks, T5 uses both. This work utilises Google’s T5 text-to-text transformer, which is initially trained for English language, however Allegro research team [3] retrained this model for Polish language and shared final models. Training this model for a new language is very time and resource consuming, but thanks to transfer learning technique it is enough to perform this only once, and afterwards fine tune it for a specific downstream task. The small and base versions of T5 model were used in experiments.

3.2 Dataset

First step required to train a model is to gather enough data of sufficiently high quality. We used datasets from both task 2 and 3 of PolEval’19 competition [4]. Task 2 data is specific to lemmatisation however it is not very big. Therefore we added tagged data from entity linking task 3. The final dataset consists of 170k pairs of orthographic and lemmatized forms. Where 20k samples were from task 2 data, 100k were phrases from task 3, and 50k were single words from tasks 3.

3.3 Additional context information

First experiments were performed only on the specified words or phrases. Then the addition of context was explored. To provide context during training each sample in data set has context attached. Maximal context span was set to 64 words before and after selected phrase. If the phrase was near the beginning of document (or end) and the context could not be fully filled the remaining space was left empty. Additionally, phrases position within context was clearly marked to provide mechanism to extract only a portion of the context.

3.4 Evaluation

To evaluate performance of the model three accuracy metrics are utilised, each counting prediction as correct only if all characters are exactly matched to the ground truth:

- AccCS - accuracy case sensitive, all characters in prediction are exactly the same
- AccCI - accuracy case insensitive, all characters in prediction are the same however, lowercase and uppercase letters are treated as equal
- Score - combination of the two, given by equation [1]

\[ \text{Score} = 0.2 \times \text{AccCS} + 0.8 \times \text{AccCI} \] (1)

2http://2019.poleval.pl/index.php/tasks/task2
3https://ml.allegro.tech/
4http://2019.poleval.pl/
4 Experiments

4.1 Training with no context

T5 model was retrained with different configurations to perform lemmatisation task. In first experiment context was not used, only the phrases that were to be lemmatized. That solution is consistent with the previous approaches. Final results are summarized in table 1. Baseline configuration did not perform any computing, only passed input as an output, accuracy score around 50% indicates that half of the input is already in its base form, and the model needs to recognise that no changes are required. Our text-to-text solution outperforms previous results significantly.

4.2 Fixed context

Next experiment utilised additional context information. The hypothesis was that given more information about context in which the phrase was used should allow the network to make better predictions [5], [3]. Both the phrase and the context are given at the input of the network, where first comes the phrase, then separation token and context, filled with padding tokens to match required input size. Table 2 summarizes the results, Additional context information improved the score by up to 1.9% for base model size and 1% for small version.

4.3 Variable context

While using fixed context size for prediction yields good results, it is not always practical to use. In many real world scenarios context size is not known in advance. For such situations we have trained a model with variable context length configuration. In training each sample had 30% probability to be provided without context, for remaining samples context length was randomly chosen from 8 to 64 range with uniform distribution. To evaluate the model a series of tests were performed: a complete test set without context, range of contexts of fixed scope, and finally a test with the same variable context settings as during training. Final results are presented in Table 1.

This series of experiments allowed as to check how context length affects prediction accuracy (fig. 1). It is visible that even small context increases accuracy significantly, and the greater the scope of the context the better results however, accuracy increase with context length is not as prominent as from no context to short context.

5 Conclusion

Presented experiments confirm the validity of text to text approach. Additional context information improved the prediction accuracy, also increase of context length correlated with the increase of accuracy however, even 8 words context was enough to rise the performance notably. Presented approach allowed to use one model regardless of context length or presence, facilitating practical use. The best model achieved 92.16% case insensitive accuracy significantly outperforming previous solutions.
Context based lemmatizer for Polish language

| Model | context | AccCS | AccCT | Score |
|-------|---------|-------|-------|-------|
| small | no      | 87.51 | 89.33 | 88.97 |
|       | fixed 8 | 88.24 | 90.00 | 89.65 |
|       | fixed 16| 88.36 | 90.11 | 89.76 |
|       | fixed 32| 88.51 | 90.22 | 89.88 |
|       | fixed 64| **88.55** | **90.24** | **89.90** |
|       | variable| 88.19 | 89.95 | 89.60 |
| base  | no      | 88.05 | 90.21 | 89.77 |
|       | fixed 8 | 90.09 | 91.92 | 91.56 |
|       | fixed 16| 90.25 | 92.01 | 91.66 |
|       | fixed 32| 90.32 | 92.10 | 91.74 |
|       | fixed 64| **90.39** | **92.16** | **91.80** |
|       | variable| 89.79 | 91.57 | 91.22 |

Table 3: Variable context

Figure 1: Lemmatizer score depending on context length, for network trained using variable context method. Dot points indicate tests with fixed context, dashed line indicate test with variable context.
Context based lemmatizer for Polish language

All models were trained on PLGrid infrastructure using Prometheus servers with NVIDIA V100 GPUs. Time required to train one epoch varied depending on model size, from 0.5h for small, 2.5h for base, to over 9h for large. Each configuration used 4 GPUs.

Acknowledgments

This research was supported in part by PLGrid Infrastructure.

References

[1] Abhisek Chakrabarty, Onkar Arun Pandit, and Utpal Garain. Context sensitive lemmatization using two successive bidirectional gated recurrent networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Vancouver, Canada, 1:1481–1491, 2017.

[2] Milan Straka, Jan Hajic, and Jana Strakova. Udpipe: Trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), 2016.

[3] Jenna Kanerva, Filip Ginter, and Tapio Salakoski. Universal lemmatizer: A sequence to sequence model for lemmatizing universal dependencies treebanks. arxiv, 2019.

[4] Krzysztof Wróbel. Krnt: Polish recurrent neural network tagger. In Zygmunt Vetulani and Patrick Paroubek, editors, Proceedings of the 8th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, pages 386–391. Fundacja Uniwersytetu im. Adama Mickiewicza w Poznaniu, 2017.

[5] Toms Bergmanis, Katharina Kann, and Hinrich Schutze. Training data augmentation for low-resource morphological inflection. Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection, page 31–39, 2017.

[6] Toms Bergmanis and Sharon Goldwater. Context sensitive neural lemmatization with lematrus. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). Association for Computational Linguistics, New Orleans, Louisiana, 1:1391–1400, 2018.

[7] Thomas Muller, Ryan Cotterell, Alexander Fraser, and Heinrich Schutze. Joint lemmatization and morphological tagging with lemming. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Lisbon, Portugal, page 2268–2274, 2015.

[8] Milan Straka. Conll 2018 shared task- udpipe baseline models and supplementary materials. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (UFAL), Faculty of Mathematics and Physics, Charles University., 2018.

[9] Daniel Kondratyuk, Tomas Gavenciak, Milan Straka, and Jan Hajic. Lemmatag: Jointly tagging and lemmatizing for morphologically rich languages with brnns. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, page 4921–4928, 2015.

[10] Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Geraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kubler, David Yarowsky, Jason Eisner, and Mans Hulden. Conll-signormorphon 2017 shared task: Universal morphological reinflection in 52 languages. In Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection. Association for Computational Linguistics, pages 1–30, 2017.

[11] Katharina Kann and Hinrich Schutze. The lmu system for the conll-sigmorphon 2017 shared task on universal morphological reinflection. In Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection. Association for Computational Linguistics, pages 40–48, 2017.

[12] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140):1–67, 2020.