NLP2 Project: *Cross-Lingual Transfer Learning*

Teaching assistant: Dennis Ulmer
dennis.ulmer@mailbox.org

April 20, 2020

1 Introduction

1.1 Motivation

Although Deep Learning has enabled tremendous progress in Natural Language Processing within the last decade, most research in the field has focused on English.[1] Even though many works consider e.g. German, French, Arabic or Mandarin, a plethora of other smaller languages, some even with hundreds of millions of speakers, lack state-of-the-art models for tasks like dependency parsing. Many of these languages also do not possess data sets that are sufficiently big to train modern architectures from scratch. Recently, *Transfer learning* has also been established as a technique in NLP to apply pre-trained models to a target task where enough data samples are not available (Howard and Ruder, 2018). In its most general form, transfer learning describes the process of applying knowledge learned on one task to another one. This has even been shown to work where the tasks are different languages, i.e. training a model on a *source* language using large corpora and then transferring it to a *target* language with sparse resources (Eisenschlos et al., 2019; Agic et al., 2016). In the following, models that are being applied to the language they were trained on will be called *native*, and models used for a different language will be named *transferred* models.

1.2 Assignment

In this assignment, the goal is to try out one of many techniques for cross-lingual transfer learning, focusing on either of the following two:

- *Zero-Shot Cross-Lingual Transfer*: In this case, a pre-trained model is trained on a source language and then applied to a target language immediately without any fine-tuning. In order to be able to process the (unseen) target language, two semantic vector spaces - one with the word embeddings of the source and the other with the vectors of the target language are being combined, either by learning a projection from target to source or by mapping them both into a completely new, joint vector space.
- *Few-Shot Cross-Lingual Transfer*: Here, a model that has been trained with a multilingual vocabulary is being transferred to a target language and then fine-tuned.

In our case, our source models are either being trained on an English or a multilingual corpus and then transferred to Dutch. The source and the target task at hand are language modelling: In language modelling we try to assign a probability to a sequence of words or tokens. Being able to accurately model the distributions present in language is deemed to be an important achievement for other downstream tasks, which is why it is often used as the source task in a NLP transfer learning setting. In this instance, we will also use it for the target task. Dutch is fortunate enough to have a

[1] See e.g. this article analyzing 20 years of ACL publications.

state-of-the-art architecture, trained exclusively on Dutch, being publicly available (Vries et al., 2019), which allows us to contrast two types of models: One trained on English and then applied to Dutch and another one trained on Dutch from scratch. We can then observe for this specific pairing where these two types of models diverge in their behaviour during inference. Therefore, the last step of the project lies in a comparative analysis.

While some pointers to publicly available models and relevant methodology will be provided in section 4, the project will leave a lot of freedom to you. For one, you should decide whether to employ the *Zero-Shot* or the *Few-Shot* approach and how to test for differences between the models during the last step. The motivations for these decisions are expected to be explained and critically reflected on in the final report.

2 Deliverables

1. Jupyter notebook, due May 22nd. The notebook should contain the entire pipeline from data preparation to word embedding alignment / model fine-tuning and the final comparative analysis. Functions or classes are encouraged to be defined in Python files externally, as long as the main functionality is listed in the notebook.

2. Short paper, due May 22nd. The short paper should contain four pages (references excluded) and be written according to the latest ACL guidelines.[2] A suggested page distribution is as follows:

   (a) **Introduction**: Introduce the reader to your research area, summarize your contributions and highlight the relevance of your research (ca. 0.5 pages);

   (b) **Related Work**: Summarize research papers relevant for your work. Be brief, since this is a short paper (ca. 0.5 pages);

   (c) **Approach**: The content of this section depends on your project (ca. 1 page). For the *Cross-lingual Transfer Learning* project, it should include the following:

   - Motivation for choice of source model.
   - Motivation for choice and details of transfer procedure.
   - Evaluation of the transferred model and the native model in terms of test perplexity.
   - Comparison of the transferred to the native model via custom experiments.
   - In the appendix: Details about experiments and training that are relevant for reproducibility, but non-essential to understand the paper's contents (exact hyperparameters for different training runs, data processing, additional plots and figures).

   (d) **Experiments and Results**: Explain the precise experimental setup used and the numerical results your models achieved (1 page);

   (e) **Discussion**: Discuss your findings, critically reflect on weak points in your methodology and support your arguments with plots and figures where applicable. Briefly outline future work (1 page).

3. Poster presentation, due May 20th, 2020. Compress the paper's content into a single-page poster that could be presented at a conference. Support the textual content through visual aids, such as tables and graphs that facilitate fast understanding of the paper's contributions and main results.
   For the *Cross-Lingual Transfer Learning* project, the poster should contain a (potentially graphical) description of the transfer learning procedure and highlights from the experiments contrasting the two model types.

[2] The ACL template with specifications can be found here.

3 Suggested Schedule

In the following, I will lay out a suggested schedule for this project. Please keep in mind that the time allocated for the project is quite short and that some critical components - especially aligning vector spaces or fine-tuning a model - usually take up more time than expected or have to be repeated in case an error is spotted. It is therefore recommended to try to finish these parts **as early as possible** and to test / verify parts of your code and your experimental conditions before running time-consuming programs. Please make sure to have the most essential parts of your project done before going into any further experimentation.

3.1 Week 1: Literature & Preparation

During the first week, you should first familiarize yourself with the relevant literature. Furthermore, you choose one of the two transfer learning variants in this project and outline the necessary steps to get them running in practice. This includes choosing adequate models and other resources and thinking about the code that will be necessary to tie them together. After making a choice, the work should begin immediately. General sections of the paper can also already be written during the first week.

3.2 Week 2: Cross-lingual Transfer

The actual transfer-learning part constitutes the core and biggest part of the project and should therefore be made working as early as possible and finished within the second week. In parallel, the procedure can already be described within the report and ideas for the final experiments between the native and the transferred model can be brainstormed and necessary code implemented.

3.3 Week 3: Contrasting transferred and native model

Lastly, you are expected to compare the transferred and native model in different experiments during the last week. The goal here is to find out which shortcomings cross-lingual transfer learning might produce: Does it struggle with certain word types or syntactic structures? Where does it potentially even exceed the native model? What are adequate metrics to compare the models and how can we make sure that measured differences are not due to other factors? Can we come up with hypotheses to explain our observations and test them somehow?

3.4 Week 4: Deliverables

Before handing in the project, make sure that all the decisions in your project are well motivated and described in your report. Reflecting on drawbacks and limitations is also highly encouraged. The code should ideally be formatted according to the `Python PEP8`[3] guidelines and sufficiently well documented.[4] I encourage to only keep the main logic inside the notebook and import other functionalities from other modules. The poster should feature the main ideas, motivations and insights and illustrate them with suitable graphs and tables. It is recommended to reduce all bullet points to their most minimal formulations. The usage of long sentences and big blocks of text is strongly discouraged.

[3] See https://www.python.org/dev/peps/pep-0008/.

[4] I do not necessarily expect elaborate docstrings, but it should be possible to quickly fathom the functionality of a piece of code without having to disentangle complex one-liners and crpytic variable names. Descriptive variable names and short comments inside the code can help here.

4 Pointers & Resources

As mentioned before, in this project you will have to choose whether to employ a *Zero-Shot* or a *Few-Shot Cross-Lingual Transfer Learning* approach. The former will require to align the English and Dutch semantic vector spaces, either by mapping target language embeddings into the source language vector space (Mikolov, Le, and Sutskever, 2013; Zhang et al., 2016; Smith et al., 2017) or by mapping both into a shared vector space (Faruqui and Dyer, 2014), both times using a bilingual dictionary. A comprehensive overview over multilingual embeddings is given in Ruder, Vulić, and Søgaard, 2019. In the latter case, a pre-trained model with a multilingual vocabulary will be fine-tuned on the target language, i.e. Dutch. Such fine-tuning procedures for NLP are e.g. given in Howard and Ruder, 2018 and Eisenschlos et al., 2019.

The following list contains some helpful resources and links:

- An English pre-trained two-layer LSTM from Gulordava et al., 2018 is available on their repository.
- HuggingFace provides a library of several transformer-based models on `GitHub`, including BERTje (Vries et al., 2019) and a multilingual BERT.
- Pre-trained Dutch word embeddings can be taken from Polyglot.
- For language modelling, sentences will run over batch boundaries. An example of how to batch correctly can be found here.
- Sometimes `Google Scholar` only lists ~~`arXiv`~~ citations for papers although they were published at some conference or in some journal. This site can help fetch the right info for citations.

Under this Canvas link, I will already provide some resources in an archive. It contains the following files:

- `train.txt`, `valid.txt`, `test.txt`: Data splits for the transfer task, based on the Dutch Wikipedia.
- `dutch_top1k.txt`: File containing the 1000 most frequent words in Dutch.
- `en_nl_dict.txt`: Download of the `dict.cc` English-Dutch dictionary.
- `filtered_en_nl_dict`: New, filtered and clean bilingual dictionary generated using `generate_dict.py` based on the previous two files, which can be used for the zero-shot transfer.

References

Agic, Zeljko et al. (2016). "Multilingual Projection for Parsing Truly Low-Resource Languages". In: *Trans. Assoc. Comput. Linguistics* 4, pp. 301–312. URL: https://transacl.org/ojs/index.php/tacl/article/view/869.

Eisenschlos, Julian et al. (2019). "MultiFiT: Efficient Multi-lingual Language Model Fine-tuning". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 5701–5706. DOI: 10.18653/v1/D19-1572. URL: https://doi.org/10.18653/v1/D19-1572.

Faruqui, Manaal and Chris Dyer (2014). "Improving Vector Space Word Representations Using Multilingual Correlation". In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pp. 462–471. DOI: 10.3115/v1/e14-1049. URL: https://doi.org/10.3115/v1/e14-1049.

Gulordava, Kristina et al. (2018). "Colorless Green Recurrent Networks Dream Hierarchically". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pp. 1195–1205. DOI: 10.18653/v1/n18-1108. URL: https://doi.org/10.18653/v1/n18-1108.

Howard, Jeremy and Sebastian Ruder (2018). "Universal Language Model Fine-tuning for Text Classification". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pp. 328–339. DOI: 10.18653/v1/P18-1031. URL: https://www.aclweb.org/anthology/P18-1031/.

Mikolov, Tomas, Quoc V. Le, and Ilya Sutskever (2013). "Exploiting Similarities among Languages for Machine Translation". In: *CoRR* abs/1309.4168. arXiv: 1309.4168. URL: http://arxiv.org/abs/1309.4168.

Ruder, Sebastian, Ivan Vulić, and Anders Søgaard (2019). "A survey of cross-lingual word embedding models". In: *Journal of Artificial Intelligence Research* 65, pp. 569–631.

Smith, Samuel L. et al. (2017). "Offline bilingual word vectors, orthogonal transformations and the inverted softmax". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. URL: https://openreview.net/forum?id=r1Aab85gg.

Vries, Wietse de et al. (2019). "BERTje: A Dutch BERT Model". In: *CoRR* abs/1912.09582. arXiv: 1912.09582. URL: http://arxiv.org/abs/1912.09582.

Zhang, Yuan et al. (2016). "Ten Pairs to Tag - Multilingual POS Tagging via Coarse Mapping between Embeddings". In: *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pp. 1307–1317. DOI: 10.18653/v1/n16-1156. URL: https://doi.org/10.18653/v1/n16-1156.