Ricardo Lukas Jung
6227492
Empirische Sprachwissenschaft (B.A.)
Phonetik & Digital Humanities
15th Semester
s2458588@stud.uni-frankfurt.de

Bachelor Thesis

Lexicalizing a BERT Tokenizer

Building Open-End MLM for Morpho-Syntactically Similar Languages

Ricardo Lukas Jung

Date of Submission:
January 15, 2023

Text Technology Lab
Prof. Dr. Alexander Mehler
Dr. Zakharia Pourtskhvanidze

Erklärung

Hiermit bestätige ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und keine anderen Quellen oder Hilfsmittel als die in dieser Arbeit angegebenen verwendet habe.

Ort, Datum

Unterschrift

Abstract

This is the abstract: what is this about? what was done? what where the results?

Contents

List of Figures

List of Tables

List of Acronyms

BERT Bidirectional Encoders from Transformers

CL Computational Linguistics

GerParCor German Parliamentary Corpus

LM Language Model

LSTM Long Short-Term Memory

ML Machine Learning

MLM Masked Language Model

NLP Natural Language Processing

POS Part of Speech

1 Introduction

This thesis showcases the use of specific intervention in tokenization subsystems of machine learning. The intent of this thesis is to inject linguistic bias into the machine learning framework of BERT to sharpen the analytical capacities of a masked language model. In this chapter the background, intentions and scope of the thesis are covered.

1 Motivation

WHY IS THIS SUBJECT RELEVANT There is an ongoing urge in the Computational Linguistics (CL) community to understand natural language. Research in the past decades shows use of frequentist and statistical methods (such as ZITATION) to their advantage, leading to the emergence of the first machine learning (ML) models. It became apparent that these ML models are the best currently available approach to an automated understanding of natural language. The structural parallels of machine learning to human learning have often been drawn (ZITATION)) to demonstrate how similar and more importantly: how different both can be. A powerful feature of Machine Learning (ML) (as opposed to human learning) is the possibility of actively controlling the the learning parameters in a supervised environment. To test the efficiency of ML parameters a variety of tasks (ZITATION) are designed and applied. A trained model will yield performance scores based on the quality of its training, much like humans on language tests. But the automated modeling of language is not the first instance language modelling in a broader sense. Traditional linguistics (DEFINITION has produced fundamental research the prior to the discovery of ML architectures and their implementation. While generic ML frameworks seem appealing in the presumption that they require less work to reach somewhat satisfactoy results, they are far from complete or perfect. The integration of aforementioned traditional linguistic knowledge into learning processes for machine learning is the underlying motivation of this thesis. **flota FLOTA** Language learners usually build up a lexicon consisting of lexemes which they will have to analyze accurately in order to be productive in that target language. A ML model relies on a tokenizer to create such a vocabulary (**ZITATION**). It is programmed to segment tokens into subwords (if possible) and provide a vocabulary comprising all the components needed to analyze a given string. Ideally those subwords will be part of the functional vocabulary in the target language, so called morphemes **ERKLÄRUNG**. A morpheme is defined as the smallest unit carrying meaning in a language. The morphemes of a language and its generated tokenizer vocabulary rarely coincide. Typically, tokanizer vocabularies will contain a lot of noise and linguistically nonsensical segmentations or words. Following the guiding principle that **input quality is ouput quality** not only in language learning, the morpheme vocabulary is identified as the point of leverage in the upcoming section. Note: explain why i use tokens and words, they are interchangeable right? holistic, need less attention to produce satisfactory NOT JUST TO PUSH F, BUT TO FIND A VIABLE METHOD OF MORPHEMIC

2 Hypotheses

The following research questions will be formulated for testing:

HYP1: Adjustments to tokenization have significant impact the performance of a language model.

How to achieve this hypothesis?

HYP2: Providing lexical information to a tokenizer increases benchmark accuracy on MLM tasks.

How to achieve this hypothesis?

3 Scope and Structure

The following chapters are sorted into three parts. To outline the research domain, a brief summary of the current state of morphological language modeling is given. Next, german is described paying special attention to its morphological complexity and peer languages. This serves as preface to the methodology, connecting characteristically matching languages to form a pool of possible target languages.

As main part of this thesis, the methodology is layed out. It is sectioned into a theoretical part which focuses on what implements are used and the value they hold towards lexicalizing a tokenizer

What is covered and what not? What is the shape of this thesis and what order does it have?

2 Overview

1 State of the Art

describe the most recent findings on morphologically pretrained models in machine learning literature findings on POS effect on ML

2 Target Languages

Describe german (ISO639-3: deu) and its morphological state. Compare to other languages with interlinear glossing.

This section identifies target languages that share common morphological features with German. German (ISO639-3: deu) is a west-germanic language and the official language of Germany, Austria, Switzerland, Liechtenstein and Luxemburg (Glück and Rödel 2016). It is an inflectional synthetic language with approximately 130 million speakers[1]. German is largely researched and is still paid much attention to in the scientific community. The aim is not to propose yet another case study of German, but to introduce German as a surrogate to further the scope of application on similar languages. Presumably those languages behave similarly when analyzed morphologically.

2.1 Pooling similar languages

On one side, linguistic typology has come up with many useful classifications for languages. On the other, in the pursuit of reconstructing languages Indo-European studies have established a widely accepted phylogenetic model of the diachronic dependecy of Indo-European languages. Both disciplines contribute to language classicifications that are used in this subsection.

Morphological complexity is a term to describe how languages use paradigms to connect grammatical information with lexemic information (Baerman, Brown, and Corbett 2017). Mind that morphological complexity is a nominal category to describe gradients of function-to-morpheme correspondence (**QUELLE**, not a qualitative assessment. The common denominators that make languages morphologically complex are their morphological features. Those languages that use affixation, fusion, composition and derivation (among others) are all fit candidates compared to german.

A summary of morphological typology is provided in 2017, pp. 78–93).

Thus, German will be the exemplary target language for the experimental setup.

Typological findings on Indo-European languages . and by means of composition and derivation.

[1] https://de.statista.com/statistik/daten/studie/1119851/umfrage/deutschsprachige-menschen-weltweit/ Last accessed: 09.01.23

(1) My s Marko poexa-l-i avtobus-om v Peredelkino
1PL COM Marko go-PST-PL bus-INS ALL Peredelkino
‘Marko and I went to Perdelkino by bus.’

Describe what morphological complexity is.
Bearman, Brown and Corbett
Describe what similar languages exist (typological vs topological)

3 Methodoloy

in this section the whole methodoloy is covered. what do i use in this thesis, why do i use it and lastly, how? make sure the why covers methodological implications. (vergiss nicht alle pakete als quelle im Anhang)

1 Requirements

A series of tools will help to achieve lexicalized tokenization. They will be explained in this chapter along with their methodological edge.

1.1 Machine Learning Model

Bidirectional Encoders from Transformers (BERT) is a language learning transformer model designed for Natural Language Processing (NLP) tasks (Vaswani et al. 2017). Upon release it achieved higher performance scores compared to previously used Long Short-Term Memory (LSTM) models (Devlin et al. 2018). Two main model characteristics can be observed for BERT. Firstly, it is the first Language Model (LM) to implement simultaneous attention heads, allowing for bidirectional reading. The methodological implication of reading to the left and right of a token is to include more information about the language in single embeddings. Secondly, BERT introduced the (at the time novel) Masked Language Model (MLM) method for training. The method involves masking a specified amount (default 15%) of random tokens in the input sequence. Masked tokens are guessed by the model which can then update its weights according to success or failure.

The NLP community has since developed BERT and adapted it to the needs of contemporary NLP problems (roberta, germanbert, mbert CITATION). Its wide support, comparability and versatility make BERT the model of choice for this thesis. Another notable feature in BERT is the implementation of the WordPiece tokenizer module (QUELLE?). Default BERT WordPiece tokenization is predominantly heuristic by combining strings based on a precalculated score. A variety of pre-trained tokenizers are available, although they come with a caveat. Once a tokenizer is trained on a dataset it is specific to that dataset. This means the application of a tokenizer on another dataset may result in out-of-vocabulary issues and different token/subtoken distributions.

Particularly relevant to this thesis is the option to train an own tokenizer from the base module. Usually, WordPiece generates its own set of subtokens called *vocabulary*. Tokens are then WORDPIECE ALGORITHMUS ERKLÄREN By providing an algorithmically generated vocabulary to WordPiece and then training it on a new dataset the tokenization behavior is changed.

1.2 Data

explain the data that is used

1.3 Benchmark

explain olmpics

2 Implementation

Tatsächliche Anwendung der Methoden auf die Daten

2.1 Tokenizer

ESSENTIALLY DERIVING SENSIBLE SUBTOKENS TO REPRESENT LEXEMES

Generating a custom pre-training vocabulary

Algorithm for the pre-training vocabulary:

Tokenizer Training

How did I train the tokenizer, how did it go? Which problems arose? What went well?

2.2 Masked Language Model

Model implementation and parameters, runtimes?

2.3 oLMpics Benchmark

tweak des tokenizers: segmentation ist eine frage der interpretation. The Ultimately, segmentation is a matter of interpretation. As mentioned in 1.1, the default WordPiece Tokenizer lacks A linguistically informed

The field of NLP (Glück and Rödel 2016) has been expanded ever since the emergence of the language models. Natural language processing is understood as the

Algorithm 1 Generate wordmap

```
Input: $verbs = \{v : v \in C \wedge v_{POS}\}, target$        ▷ Set of single-POS lexemic tokens
Output: $maps = (map_1, \ldots, map_{|verbs|})$
  $pair = (target, v)$
  $case = \textsc{match\_ends}(pair)$        ▷ Returns if strings match in the last or first position
  $s = \textsc{shorter}(pair)$
  $l = \textsc{longer}(pair)$
  $len = \textsc{len}(l)$
  $\delta = \Delta(len - \textsc{len}(s))$

  if any ends match then
      if $\delta$ then
          if case: left match then
              for $i = 0$ to $len$ do
                  $map[i] = f : (s[i], l[i]) \mapsto s[i] == l[i])$
              end for
              Pad map from right side with 0s to match $\delta$
          end if
          if case: right match then
              for $i = 0$ to $len$ do
                  $map[i] = f : (s[i + \delta], l[i]) \mapsto s[i + \delta] == l[i])$
              end for
              Pad map from left side with 0s to match $\delta$
          end if
      else
          $map[i] = f : (s[i], l[i]) \mapsto s[i] == l[i])$
      end if
  end if
```

4 Results

1 Benchmark

How did the model Perform in the benchmark test? Report the performance for different tasks and visualize it

2 Tokenization

Show specific examples of tokenization and analyze the qualitatively (maybe quantitatively)

5 Discussion

6 Conclusion

What was done? How did it go? What went wrong? What went well? What was learned from this? What are future applications?

In almost all statistical modeling the goal is to model reality as precisely as feasible. Language models are no exception. The accuracy of a model should increase with the number of functional components of natural language being integrated into the model. This is seen in e.g. the implementation of vocabularies, just one of many attempts to automatically identify meaningful units in language. Even higher levels of language found in domains from ordinary pragmatics to scientific reasoning are sought after in language modeling. While languages are observed to change slowly over time, sometimes dropping and adding features of their inventory, computational linguistics has to keep producing models that keep up with the reality of language. The supervised tokenization in this thesis illustrates just a small part of the potential in tailored modeling.

7 Testchapter

1 Citing

Abrami et al. 2022

2 Quoting

> “This is a quote by textquote” (DeepL 2021) “This is a quote by enquote”

3 Referencing

Short reference 1.1
Long reference subsection 1.1

Bibliography

Abrami, Giuseppe, Mevlüt Bagci, Leon Hammerla, and Alexander Mehler (June 2022). "German Parliamentary Corpus (GerParCor)". In: *Proceedings of the Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 1900–1906. URL: https://aclanthology.org/2022.lrec-1.202.

Aikhenvald, Alexandra and R Dixon (2017). *The Cambridge Handbook of Linguistic Typology*. Cambridge Handbooks in Language and Linguistics. Cambridge: Cambridge University Press. DOI: 10.1017/9781316135716.

Baerman, Matthew, Dunstan Brown, and Greville G. Corbett, eds. (2017). *Morphological Complexity*. Cambridge studies in Linguistics 153. Cambridge University Press. ISBN: 1107120640. DOI: 10.1017/9781316343074. URL: www.cambridge.org/9781107120648.

DeepL (Nov. 2021). *How does deepl work?* URL: https://www.deepl.com/en/blog/how-does-deepl-work. Last accessed: 28.12.2022.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. DOI: 10.48550/ARXIV.1810.04805. URL: https://arxiv.org/abs/1810.04805.

Glück, Helmut and Michael Rödel, eds. (2016). *Metzler Lexikon Sprache*. ger. 5th ed. Springer eBook Collection. Stuttgart: J.B. Metzler, pp. 141–142. ISBN: 978-3-476-05486-9. DOI: 10.1007/978-3-476-05486-9. URL: http://dx.doi.org/10.1007/978-3-476-05486-9.

Vaswani, Ashish et al. (2017). *Attention Is All You Need*. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.