Impact of Lexical Normalization on Twitter Sentiment Analysis

Jannik Gut & Manuel Meinen & Robin Burkhard & Bernhard Walser
Group: bubblesort
Department of Computer Science, ETH Zurich, Switzerland

***Abstract*—In this paper we show the impact of lexical normalization on the performance of different models for the task of Twitter sentiment analysis. We investigated BERT and ALBERT models of various sizes and performed lexical normalization using MoNoise in the default as well as the bad-speller mode. Our findings suggest that the impact of lexical normalization depends on the model architecture as well as the model size and that performing lexical normalization can also hurt performance. It is therefore not possible to give a final recommendation on whether it is advisable to perform lexical normalization prior to performing further data analysis.**

I. Introduction

Twitter sentiment analysis can provide insight on how the public feels in general or more specifically about a certain subject such as Covid-19 [1]. This information can be used as a base for political or financial decision making.

In the recent years there has been great progress in the field of Natural Language Processing with the advent of large, pre-trained language representation models [2] [3] [4], which augmented the performance tasks achieve on normalized data significantly. The transition to non-standard Twitter language remains a challenge, which is subject to the scientific field of lexical normalization.

Our key contributions are:

- We show the impact of lexical normalization on model performance for different model architectures.
- We investigate the use of additional training data from a slightly different data distribution.
- We provide an easy to use repository[1] for reproducing the results for further research.

II. Models and Methods

A. Model Architecture

In recent years many new language representation models were introduced. Starting with the development of ELMo in 2018 [5], which was based on a bidirectional LSTM architecture, followed by BERT [2] in 2018, which made use of the newly introduced transformer architecture with attention [6] and provided a deeper bidirectionality than ELMo.

Since then many different slight adaptations of the BERT architecture [7] [8] were introduced, such as ALBERT [3], which we also used as a second baseline next to BERT.

BERT. It is a powerful language representation model and from the official BERT GitHub repository[2] one can download different checkpoints of the model, which was pre-trained for a long period of time on the large, normalized Wikipedia Corpus and BookCorpus. The pre-trained models consist of strong representations of different words in various contexts and we can make use of that knowledge by adding a small task specific head to this BERT backbone, which allows us to classify data and fine-tune the model for our target domain [2].

ALBERT. It introduces two parameter-reduction techniques to lower memory consumption and increase the training speed of BERT [3]. They use a factorization of the embedding parameters, decomposing them into two smaller matrices. This unties the WordPiece input embedding size from the hidden layer size, which allows for a more efficient usage of the total model parameters. ALBERT shares all parameters across layers, which further improves parameter efficiency [3]. With the help of these reductions it is possible to run a bigger model on the same restricted environment, such as Google Colab[3] that was used for this project.

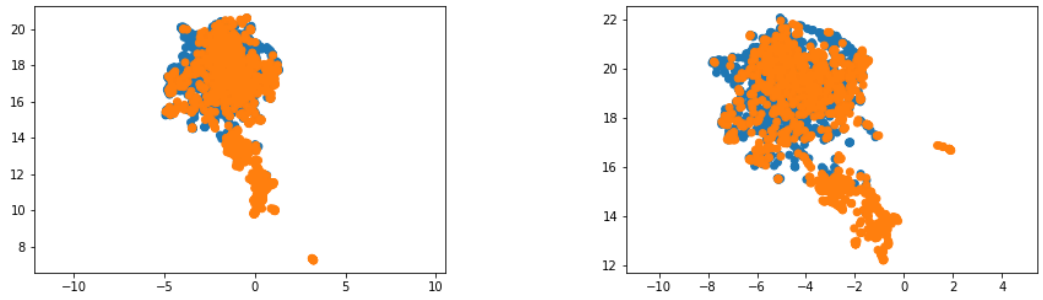
B. Datasets

Competition Data. Various people have crawled some of the millions of unlabeled tweets that get sent every day and labeled the data for various NLP-tasks, like for sentiment analysis and the competition dataset. This dataset contains about 2.5 million tweets labeled as positive or negative. The classes are balanced and the length of the tweets is visualized in Figure 2. The data is all lower-case, users and URLs are substituted by placeholders and smileys/emojis are tokenized together as one token as opposed to punctuation, which is one token each. The data is not trivially classifiable in general, at least not with Universal Sentence Encoder embeddings [9], as can be seen in the UMAP [10] Figure 1, but it is apparent, that some negative tweets are different to the big cluster consisting of mostly positive but still a lot of negative samples. These samples may come from different languages or non-understandable samples, which are mostly classified as negative.

[1] https://github.com/berniwal/CIL_Project

[2] https://github.com/google-research/bert

[3] colab.research.google.com/

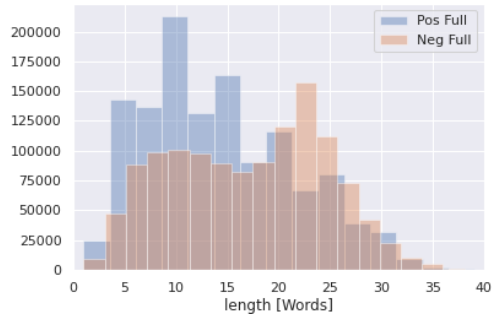


(a) original data (b) normalized data

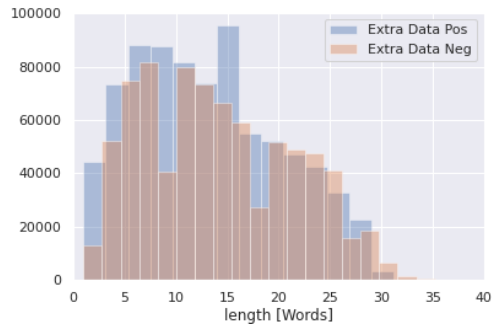
Figure 1: UMAP [10] visualization of Universal Sentence Encoding embeddings [9] of the first 1000 samples of each label with and without lexical normalization. Negative samples are orange, positive samples are blue.

Additional Training Data. As it is widely known that more good data most likely leads to better performance, it makes sense to make use of external data. This data most likely comes from a different data distribution because it was collected in a different period of time or has been pre-processed or labeled slightly differently. The difference in our case becomes apparent when looking at the length of the tweets in Figure 2. But this data should still be a lot closer to the target data than the normalized data with which the language models were pre-trained. We include one additional dataset with 1.6 million tweets [11], which is already perfectly balanced. But URLs and usernames still had to be replaced with placeholders. Also the labels had to be adjusted and some meta-data had to be deleted, as it was not provided with the competition data.

When using additional data, we first train our model with this extra data. This should give the model the possibility to extract important additional information and lead it in the right direction for the further training on the target data as can be seen in Section III-A.



(a) original data



(b) extra data

Figure 2: Histogram showing the different distributions of tweet length in words between labels.

C. Lexical Normalization

Lexical normalization is the task of translating a text from a non-canonical domain (Twitter) into a more canonical one (standard English). This seems reasonable because pre-trained models like BERT & ALBERT are trained on standardized texts and therefore should work best on standardized vocabulary and tokenization (word splitting). We use MoNoise [12], which is based on candidate generation and candidate ranking. The former will generate different candidates for the original word, the latter ranks this candidate list and tries to elect the correct candidate.

Candidate Generation. It consists of multiple complementary modules, which all handle different types of anomalies. Therefore, the candidate list contains the following: the ORIGINAL TOKEN, the closest 40 words in a SKIP-GRAM MODEL [13] based on the cosine distance, words coming from the ASPELL SPELL CHECKER [14], based on the character edit distance and the phonetic distance and CANDIDATES COMING FROM A LOOK-UP LIST, learned from training data. Further, to capture abbreviated words, WORDS THAT START WITH THE SAME CHARACTER SEQUENCE as the original word and can be found in the Aspell dictionary are also added and lastly, WORD SPLITS which were generated by splitting the word in every possible

position and check if the resulting words are canonical[12].

Candidate Ranking. It computes a score for every candidate based on a random forest classifier with the following criteria: if the candidate is the ORIGINAL TOKEN, if it is generated by the abbreviated words MODULE, if it can be FOUND IN THE ASPELL DICTIONARY, if the characters still occur in the same ORDER, if the token contains any ALPHABETICAL CHARACTERS, COSINE DISTANCE BETWEEN THE CANDIDATE AND THE ORIGINAL WORD in the vector space, the ranking and lexical/phonetical EDIT DISTANCES, the count of the OCCURRENCES OF THE CORRECTION PAIR in the lookup-list, the UNIGRAM AND BIGRAM PROBABILITY with the previous word, the bigram probability with the next word and the LENGTH of the original word as well as the candidate word [12].

Bad-Speller Mode. A different mode of the Aspell [14] spell checker that is also evaluated where the model is more tailored for bad spellings rather than having a balance between typos and true-misspellings. This mode never performs typo-analysis and allows for candidates with a larger distance to the original word, but therefore is much slower [12].

Example. The tweet "new pix comming tmr. thx for the suport!" gets changed to:

- "new pix coming tmr . thanks for the support !" (MoNoise)
- "new pix coming tomorrow . thanks for the support !" (MoNoise with Bad-Speller)

D. Implementation Details

For this project the bert-for-tf2[4] implementation of BERT was used and their example notebook for sentiment analysis on the IMDB Movie Review dataset served as starting point.

The models were trained over multiple epochs with the categorical cross-entropy loss and Adam optimizer. The batch-size was 32 and a reduce on plateau learning rate scheduler with a patience of 5, minimum learning rate of 1e-7 and start learning rate of 1e-5, 1e-6 for BERT and ALBERT respectively was used. Training was stopped early at the minimum learning rate if the patience threshold was reached again. Validation was issued every 10 steps, whereas every step consisted of roughly 11.9k samples each.

Training was performed on multiple TPUs, which are freely provided through Google Colaboratory[5].

III. RESULTS

In the following section we present the results of an ablation study for different BERT/ALBERT model architectures on the use of additional data and lexical normalization. Results from the Kaggle public competition leaderboards are shown in Table I and Table II.

[4]https://github.com/kpe/bert-for-tf2

[5]https://colab.research.google.com/

Model	Accuracy
BERT-base	0.89580
BERT-large	0.89240
ALBERT-base	0.87380
ALBERT-large	0.87540
ALBERT-xlarge	**0.90040**
ALBERT-xxlarge	0.89740

Table I: Public leaderboard accuracies of different models without lexical normalization or additional data.

A. Model Comparison

Table I shows that there is hardly any significant difference between BERT-base and BERT-large. ALBERT-xlarge and ALBERT-xxlarge are both better and achieve around 90% accuracy. Although the highest accuracy was achieved by ALBERT-xlarge without any lexical normalization or additional data, we decided to present the ablation study on ALBERT-xxlarge as it showed better end results and therefore includes our submitted model.

B. Additional Training Data

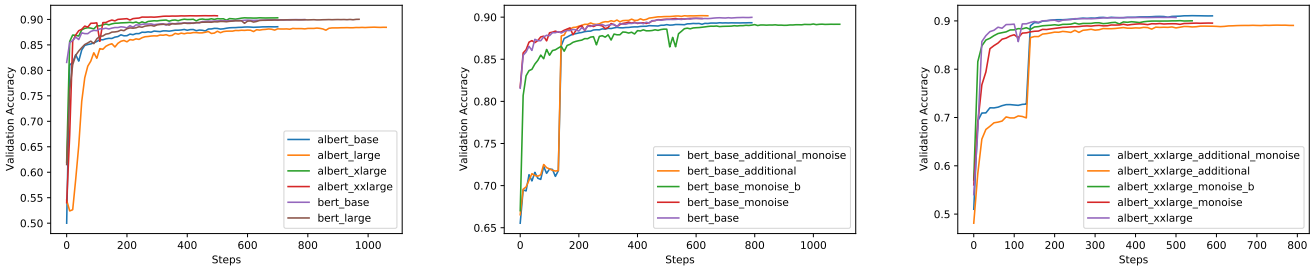
With additional training data we can see a slight decrease in performance. Even though the model was clearly able to extract some useful information from the additional data first, reaching around 70 percent accuracy on the target distribution in Figure 3, according results can also be observed by looking at Figures 4-8 in the appendix, it did not help increasing the overall accuracy.

C. Lexical Normalization

As can be seen in Table II both modes of lexical normalization seem to decrease the accuracy on the public leaderboard, while keeping the same BERT-base or ALBERT-xxlarge model. The difference between the softer MoNoise mode and the more aggressive bad-speller mode is only marginal as less than 150k of the 2.5 million tweets have been affected by the bad-speller mode.

	BERT-base	ALBERT-xxlarge
Normal	0.89580	0.89740
MoNoise	0.88720	0.88900
Bad-Speller	0.88560	0.89080
Add. Data	0.89440	0.87720
MoNoise + Add. Data	0.8848	**0.90360**

Table II: Public leaderboard accuracies of the BERT-base/ALBERT-xxlarge models using neither additional data nor lexical normalization (Normal), using MoNoise lexical normalization in the default (MoNoise) or Bad-Speller mode (Bad-Speller), using additional data (Add. Data) or using additional data and default MoNoise combined (MoNoise + Add. Data).



(a) Base Models (b) Ablation BERT (c) Ablation ALBERT

Figure 3: Validation accuracies, on the left on different models without lexical normalisation or additional data, in the middle on BERT-base with different input data and on the right the ALBERT-xxlarge model with different input data.

IV. Discussion

As can surprisingly be seen in Table II, the extra step of making lexical normalization has a negative effect on the accuracy, even though usually BERT works best if it can work on the well-written vocabulary it was trained on. A possible explanation for this might be, that BERT picked up the slang and used the bad-spelling of words as another feature, which is a better indicator than the deeper understanding of a well-written word.

We assume that the model was able to extract some general features but it seems that there is no extra knowledge extracted from the additional data which the training data not already provides.

However, the best model found uses both lexical normalization without bad-speller and the additional data on the biggest architecture tested (ALBERT-xxlarge). This model is also the only one that shows improvements over the baseline using a combination of the two methods. As the margins are relatively small, it might be that this model can match the variance of the data best.

V. Summary

Lexical normalization sometimes helps to increase the performance of NLP tasks in different domains, however for this task of sentiment analysis on a Twitter dataset, there might be information in the unnormalized text which greatly benefits the model and outweighs the influence of lexical normalization.

References

[1] Martin Müller, Marcel Salathé, and Per E Kummervold. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter, 2020.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[3] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2019.

[4] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2019.

[5] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations, 2018.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[7] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[8] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp, 2019.

[9] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder, 2018.

[10] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*, February 2018.

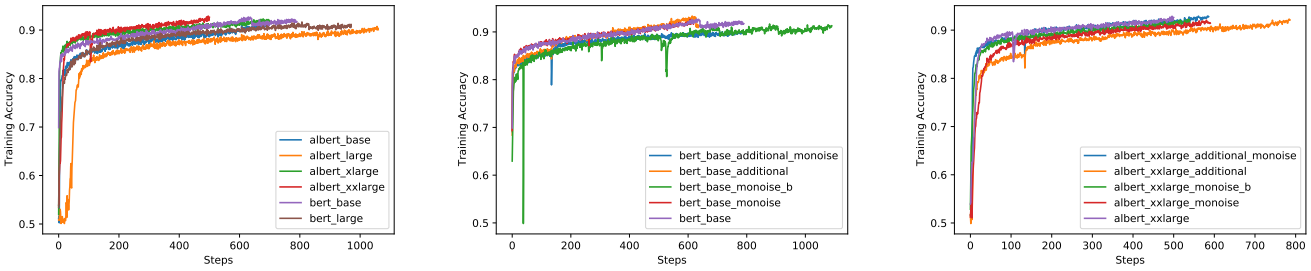
[11] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision, 2009.

[12] Rob van der Goot and Gertjan van Noord. Monoise: Modeling noise using a modular normalization system, 2017.

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

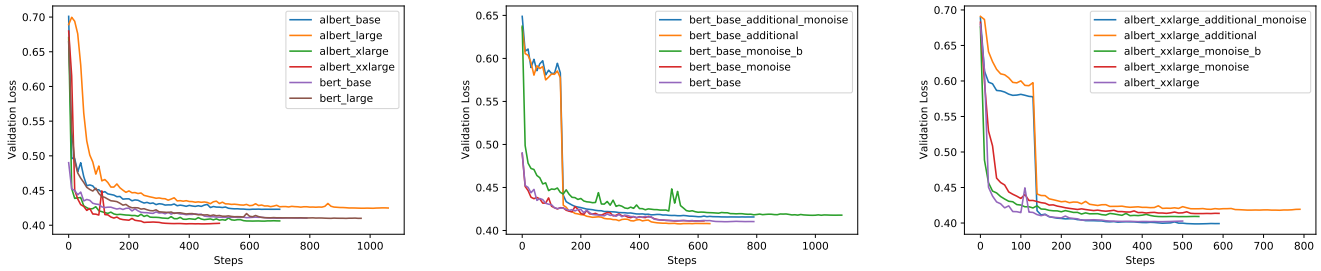
[14] Aspell. *Aspell Documentation*, (accessed July 6, 2020).

APPENDIX



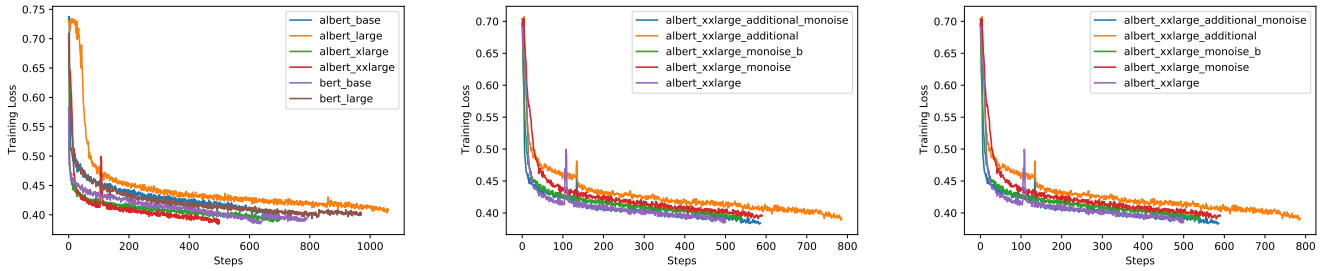
(a) Base Models (b) Ablation BERT (c) Ablation ALBERT

Figure 4: Training accuracies, on the left on different models without lexical normalisation or additional data, in the middle on BERT-base with different input data and on the right the ALBERT-xxlarge model with different input data.



(a) Base Models (b) Ablation BERT (c) Ablation ALBERT

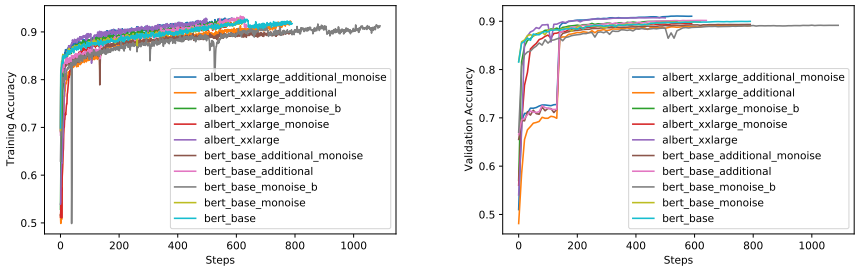
Figure 5: Validation losses, on the left on different models without lexical normalisation or additional data, in the middle on BERT-base with different input data and on the right the ALBERT-xxlarge model with different input data.



(a) Base Models (b) Ablation BERT (c) Ablation ALBERT

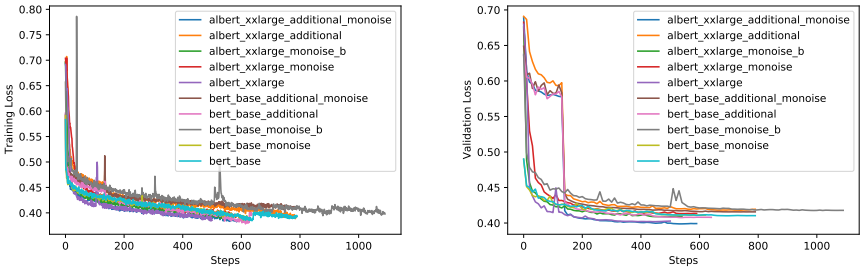
Figure 6: Training losses, on the left on different models without lexical normalisation or additional data, in the middle on BERT-base with different input data and on the right the ALBERT-xxlarge model with different input data.

APPENDIX



(a) Training (b) Validation

Figure 7: Accuracies, on the left on different models without lexical normalisation or additional data, in the middle on BERT-base with different input data and on the right the ALBERT-xxlarge model with different input data.



(a) Training (b) Validation

Figure 8: Losses, on the left on different models without lexical normalisation or additional data, in the middle on BERT-base with different input data and on the right the ALBERT-xxlarge model with different input data.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Impact of Lexical Normalization on Twitter Sentiment Analysis

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Burkhard	Robin
Gut	Jannik
Meinen	Manuel
Walser	Bernhard

With my signature I confirm that

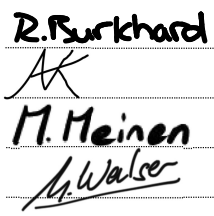
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date

Zurich, 31.07.2020

Signature(s)



For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.