Comparing the performance of various Swedish BERT models for classification

Daniel Holmer, Arne Jönsson
Deptartment of Computer and Information Science
Linköping University
danho775@student.liu.se|arne.jonsson@liu.se

Abstract

We present results from fine-tuning BERT using different models, one multilingual, and two dedicated Swedish BERT models, for the task of classifying Swedish texts as of either easy-to-read or standard complexity in their respective domains. The performance on the text classification task using the different models is then used to compare the BERT models with each other and with feature representation methods used in earlier studies. The results show that all models performed better on the classification task than the previous methods of feature representation. Furthermore, the dedicated Swedish models show better performance than the multilingual model, with the Swedish model pre-trained on more diverse data outperforming the other.

1 Introduction

One of the simplest ways to represent text as fetures is to use bag-of-words (BOW), where each word in the text is stored together with their relative frequency, ignoring word position. A more advanced way to represent features is by using word embeddings. One popular way to create these word embeddings is word2vec (Mikolov et al., 2013). Word2vec can capture the meaning of words, providing more competent feature representations than the previously mentioned BOW approach. The embeddings from word2vec work relatively well, but can not capture different contextual meanings of words. This will significantly limit the quality of the feature representation since, in human natural language, the context of a word is of great importance to its meaning. To be able to account for context is computationally expensive, but during the evolution of more and more intricate deep-learning network architectures, new methods capable of creating context-dependent word embeddings have emerged.

One of these newer and potentially more powerful methods is BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019). BERT builds upon the multi-layered bidirectional Transformer encoder described in Vaswani et al. (2017), which allows the model to learn bidirectional representations of unlabeled texts. That is, the model learns contextual relations between all the words in a sequence, providing a rich feature representation for every word. To learn these representations, a BERT model is pre-trained on a large number of text documents. The pre-trained model can subsequently be fine-tuned to perform many specific tasks, just by changing the final output layer. This form of transfer learning has shown to be an efficient way to apply general language representations to a specific domain, without having to learn the entire model from scratch.

It is tempting to believe that every new, theoretically more advanced feature representation method is always going to outperform the theoretically simpler ones. In reality, this is not always the case. For example, Santini et al. (2019a) found that the previously mentioned BOW produced better results (in the form of higher weighted average f-scores) on the task of classifying Swedish texts as of low or standard complexity, than the on paper superior word2vec method. Santini et al. (2019a) also notes that this unstable behavior of word2vec has been observed in other domains (Wendlandt et al., 2018). The fact that bigger is not always better makes it important to study BERT in different domains, and determine if similar problems exist.

2 Datasets

Three Swedish datasets were used in this study; the eCare and DigInclude subsets in Santini et al. (2019a), as well as an extended version of the DigInclude subset, see Table 1.

Table 1: Summary of datasets

	eCare	DigInclude sentences	DigInclude documents
Domain	Medical	Authorities	Authorities
Unit of analysis	Document	Sentence	Document
Entries (total)	3423	17,502	6,164
Entries per class (easy)	863	3,827	3,082
Entries per class (standard)	2,560	13,675	3,082
Class balance (easy/standard)%	25/75	22/78	50/50
Number of words	424,278	233,094	530,089

The first dataset used is a subset of the eCare corpus (Santini et al., 2019b), which consists of 3423[1] web pages of texts regarding chronic diseases, labeled as either `lay` or `specialized` by a native lay speaker. The `specialized` pages were judged to contain medical jargon. In contrast, the `lay` pages were judged to contain language understandable to a person with no medical training, and is seen as an easy to read version of the standard language (medical jargon) used in the domain. The distribution between the two categories are: 2560 `specialized` pages (75%), and 863 `lay` pages (25%). The total amount of words in the whole eCare dataset is 424,278.

The second dataset used is a subset of the DigInclude corpus (Rennes and Jönsson, 2016). The dataset consists of a total of 17,502 sentences crawled from Swedish authorities' web sites. 3,827 (22%) of them were categorized as easy-to-read (`simple`) and 13,675 (78%) were categorized as of `standard` complexity. There are a total number of 233,094 words in the subset. Both of these datasets from Santini et al. (2019a) are unbalanced, with one class clearly outweighing the other.

The third dataset used is another subset of the DigInclude corpus (Rennes and Jönsson, 2016). However, instead of single sentences, each entry consists of several sentences. This is a balanced dataset of 6,164 entries, with 3,082 (50%) entries classified as easy-to-read (`simple`) and 3,082 (50%) entries classified as of `standard` complexity. The total number of words are 530,089.

3 BERT Models

We used three pre-trained BERT models, one multilingual and two Swedish, see Table 2.

The multilingual model released alongside the original BERT paper (Devlin et al., 2019) is trained on the 104 languages with the largest Wikipedias. All of the Wikipedia articles for each language were used as the training data. Since there is a large imbalance between the number of articles for different languages, the less common languages are over-sampled, and the more common languages are under-sampled. This method normalized the dataset somewhat, but there is still a fairly large bias towards the most common languages.

The first Swedish pre-trained BERT model used in this study was created by KBLab at Kungliga biblioteket (the National Library of Sweden) (Malmsten et al., 2020). The model was pre-trained from approximately 15-20GB text (200M sentences, 3000M tokens) from various sources, including books, news, government publications, Swedish Wikipedia, and internet forums. The second Swedish pre-trained BERT model was developed by Arbetsförmedlingen (the Swedish Public Employment Service)[2]. The model was trained on around 2M articles, with 300M words retrieved from Swedish Wikipedia. This model was trained on a significantly smaller dataset than the multilingual and the KBLab model. Also, the model is uncased, which means that the model processes every word as lower case WordPiece tokens.

4 Method

To implement the BERT architecture, the Python library *HuggingFace's Transformers* (HF Transformers) (Wolf et al., 2019) was used. Due to memory limitations, the maximum length for the input sequence is 512 tokens. Since BERT expects that all inputs are of a fixed length, any document shorter than the chosen sequence length must be padded with the special `[PAD]` token.

However, the limited max sequence length of 512 tokens is still very expensive computationally. As a result, in order to use the sequence length 512 with the GPU provided by Google Colab[3], the

[1] For reasons explained in Section 4, the class balance and the number of entries has been altered from the subset used in (Santini et al., 2019a).

[2] https://github.com/af-ai-center/SweBERT

[3] Nvidia Tesla P100 with 16GB RAM

Table 2: BERT models overview

	Multilingual	**Swedish**	
	BERT-Base	KB	AF AI
Encoder layers (blocks)	12	12	12
Feed-forward network hidden layer size	768	768	768
Attention heads	12	12	12
WordPiece casing	Cased	Cased	Uncased
Vocabulary size	119,547	50,325	30,522
Training data (approx.)	55-60GB	15-20GB	5-10GB

Table 3: ZeroR baselines, breakdown

	Class	k	Acc(%)	Err(%)	P	R	F	ROC
eCare Subset (3423 webpages)	lay (863 webpages)	0.00	75	25	0.00	0.00	0.00	0.500
	specialized (2560 webpages)				0.75	1.00	0.86	0.500
	AvgF						**0.64**	
DigInclude Sentences Subset (17,502 sentences)	simplified (3,827 sentences)	0.00	78	22	0.00	0.00	0.00	0.500
	specialized (13,675 sentences)				0.78	1.00	0.87	0.500
	AvgF						**0.68**	
DigInclude Documents Subset (6,164 documents)	simplified (3,082 documents)	0.00	50	50	0.00	0.00	0.00	0.500
	standard (3,082 documents)				0.50	1.00	0.50	0.500
	AvgF						**0.50**	

maximum batch size during training would have to be reduced beyond reasonable limits to avoid memory issues. It would ultimately harm the models' performance. Due to this limitation in available hardware, the maximum sequence length was set to 128 tokens[4] which during training allowed for a batch size of 32.

Since the DigInclude sentence dataset contains sentences all shorter than 126 tokens, no additional action was taken to fit them in the maximum sequence length. On the other hand, the eCare dataset is not based on sentences; the unit of analysis is documents (formed from web pages). These consist of numerous sentences, often combining for way more than 126 tokens. The web pages of the eCare dataset were therefore split into chunks of a maximum of 126 tokens each to be able to fit the input window. The DigInclude documents dataset was also in the document format, but with lengths closer to the 126 token limit. Since the number of entries in this dataset were much larger than the eCare dataset, the cost of discarding superfluous tokens was substantially smaller. Therefore, no splitting into chunks was performed.

Each dataset was tokenized with the help of the `AutoTokenizer` class, and each model was subsequently fine-tuned in all layers with the `BertForSequenceClassification` class, both included in the HF Transformers library. For fine-tuning we used the default PyTorch cross-entropy loss function utilized by HF Transformers together with the hyperparameters: `batch size=32`, `learning rate=2e-5`, and `epochs=4`. These hyperparameters gave the best average results over all datasets, and also align with the hyperparameters recommended by Devlin et al. (2019).

[4]Which fits 126 tokens of text, as well as the obligatory [CLS] and [SEP] tokens

5 Results

Table 3 shows an overview of the ZeroR baselines. For all datasets, the *k-statistic* and *ROC* were at their base values, *0.00* and *0.500*. The baseline *AvgF* were for the eCare dataset *0.64*, for the Diginclude Sentences dataset *0.68*, and for the DigInclude Documents dataset *0.50*.

Table 4 displays the results of the implementation of the three different BERT models on the eCare dataset. All models showed somewhat similar results, but both Swedish models scored slightly higher than the multilingual one. The KB model showed *AvgF = 0.87*, with *ROC = 0.931*, and the *k-statistic = 0.66*. On the other hand, AF AI showed a higher *AvgF, (0.89)*, but a lower *ROC (0.923)*. The k-statistic for KB were *0.66*, and for AF AI *0.67*. The multilingual model scored lower on all the main statistics, with *AvgF = 0.86*, *ROC = 0.912*, and *k-statistic = 0.64*.

Table 5 depicts the results from the first DigInclude dataset, consisting of single sentences. The multilingual and the AF AI model showed similar results with *AvgF = 0.75*, while the KB model had *AvgF = 0.78*. The KB model also gave the best results on ROC (0.766) and the k-statistic (0.33), followed by the multilingual models *ROC = 0.727*,

Table 4: BERT, eCare

BERT: eCare Subset							
Multilingual	k	Acc(%)	Err(%)	P	R	F	ROC
lay	0.64	85	15	0.68	0.80	0.73	0.912
specialized				0.93	0.87	0.90	0.912
AvgF						**0.86**	
KB (Swedish)	k	Acc(%)	Err(%)	P	R	F	ROC
lay	0.66	87	13	0.71	0.81	0.75	0.931
specialized				0.93	0.89	0.91	0.931
AvgF						**0.87**	
AF AI (Swedish)	k	Acc(%)	Err(%)	P	R	F	ROC
lay	0.67	89	11	0.82	0.72	0.77	0.923
specialized				0.91	0.95	0.93	0.923
AvgF						**0.89**	

Table 5: BERT, DigInclude sentences

DigInclude Sentence Subset							
Multilingual	k	Acc(%)	Err(%)	P	R	F	ROC
simplified	0.29	75	25	0.43	0.49	0.46	0.727
standard				0.85	0.82	0.83	0.727
AvgF						**0.75**	
KB (Swedish)	k	Acc(%)	Err(%)	P	R	F	ROC
simplified	0.33	78	22	0.50	0.45	0.47	0.766
standard				0.85	0.87	0.86	0.766
AvgF						**0.78**	
AF AI (Swedish)	k	Acc(%)	Err(%)	P	R	F	ROC
simplified	0.26	76	24	0.43	0.40	0.41	0.713
standard				0.84	0.86	0.85	0.713
AvgF						**0.75**	

and *k-statistc = 0.29*. Lowest scores was shown by the AF AI model, with *ROC = 0.713*, and *k-statistic = 0.26*.

For the second DigInclude dataset, the results are displayed in Table 6. The KB model consistently showed the best results with *AvgF = 0.94*, *ROC = 0.983* and *k-statistic = 0.88*. The AF AI model scored all the second best results with *AvgF = 0.92*, *ROC = 0.976* and *k-statistic = 0.84*. Of the three models, the multilingual one performed the worst on this dataset, with *AvgF = 0.90*, *ROC = 0.975* (only slightly lower than AF AI), and the *k-statistic = 0.81*.

Finally, it is worth noting that all the classifiers performed better in all aspects on the DigInclude documents subset, compared to the sentence subset, with all the three models.

Table 6: BERT, DigInclude documents

DigInclude Documents Subset							
Multilingual	k	Acc(%)	Err(%)	P	R	F	ROC
simplified	0.81	90	10	0.86	0.97	0.91	0.975
standard				0.96	0.84	0.90	0.975
AvgF						**0.90**	
KB (Swedish)	k	Acc(%)	Err(%)	P	R	F	ROC
simplified	0.88	94	6	0.92	0.97	0.94	0.983
standard				0.96	0.92	0.94	0.983
AvgF						**0.94**	
AF AI (Swedish)	k	Acc(%)	Err(%)	P	R	F	ROC
simplified	0.84	92	8	0.90	0.94	0.92	0.976
standard				0.94	0.90	0.92	0.976
AvgF						**0.92**	

6 Discussion

It can be noted that all of the three BERT models (Multilingual, KB, and AF AI) used in this study were able to produce better results than BOW+SMO used in Santini et al. (2019a), and much better results than the ZeroR baselines (see Table 3), on the text classification task. This was shown on both the eCare and DigInclude sentence datasets, but the increase in performance was clearer on the eCare dataset than the DigInclude sentence dataset. Furthermore, the BERT models performed significantly better when classifying documents from the DigInclude document subset, compared to sentences from the DigInclude sentence subset.

The main difference in the DigInclude sentence dataset compared to the other two is that it consists of sentences, not documents. As a consequence, each entry is substantially shorter. The sentence dataset is also smaller and not balanced, and it seems that this is causing problems for the BERT models, mainly expressed in the form of a poor classification performance on the minor class. This behavior was also seen in the feature representation methods in Santini et al. (2019a). Even though the issues are not as pronounced with the BERT models used in this study as with the methods in the previous study, it is clear that they are still present. However, it is uncertain if it is the entry length, total dataset size, or the imbalance between the classes that have the most influence over this behavior in the BERT models.

7 Conclusion

The results of this study have shown that the Swedish models performed better overall than the multilingual one; the KB model scored consistently higher on all datasets, while the AF AI model scored higher on two of the datasets, and slightly lower on the third. Of the two Swedish models, the one from KB was noticeably better on two of the datasets, and on par with the AF AI model on one of the datasets.

Using BERT for Swedish text classification, undeniably shows promising results. However, the structure of the text domain seems to have a great impact on the classification results. Future studies could further investigate this relation by interpreting what structural aspects of the Swedish language the BERT models learn.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with Words at the National Library of Sweden – Making a Swedish BERT. *arXiv preprint arXiv:1910.03771*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, pages 3111–3119.

Evelina Rennes and Arne Jönsson. 2016. Towards a corpus of easy to read authority web texts. In *Proceedings of the Sixth Swedish Language Technology Conference (SLTC2016), Umeå, Sweden*.

Marina Santini, Benjamin Danielsson, and Arne Jönsson. 2019a. Comparing the Performance of Feature Representations for the Categorization of the Easy-to-Read Variety vs Standard Language. *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 105–114.

Marina Santini, Arne Jönsson, Wiktor Strandqvist, Gustav Cederblad, Mikael Nyström, Marjan Alirezaie, Leili Lind, Eva Blomqvist, Maria Lindén, and Annica Kristoffersson. 2019b. Designing an extensible domain-specific web corpus for "layfication": A case study in ecare at home. *Cyber-Physical Systems for Social Applications*, pages 98–155.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Laura Wendlandt, Jonathan K. Kummerfeld, and Rada Mihalcea. 2018. Factors influencing the surprising instability of word embeddings. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2092–2102.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.