Targeted Sentiment Analysis Using a Fine-Grained Dataset: NoReCfine

Arthur Dujardin
Department of Informatics,
University of Oslo

Lotte Boerboom
Department of Informatics,
University of Oslo

Silvia N.W. Hertzberg
Department of Informatics,
University of Oslo

Abstract

Targeted sentiment analysis (TSA) is a variant of normal sentiment analysis, with which you can analyse the sentiment towards a specific target within a text. The goal of this project was to train different neural systems to perform TSA for Norwegian text. Different design methods were examined to find out what their effects were on the recently released data set $NoReC_{fine}$. The performance of bidirectional LSTMs, GRUs and a BERT model were compared to each other and a baseline. In addition a small error analysis was performed. The results show that BERT does not outperform GRUs or LSTMs. The code has been implemented in *PyTorch* and attention was paid to the creation of online documentation on *readthedocs* [1].

1 Introduction

Sentiment analysis (SA) is a well-researched topic in Natural Language Processing (NLP). Conventionally the analysis obtains and classifies emotions or sentiments in sentences or documents as positive or negative. In this regard, there has been a parallel growth of SA identified with the growth of social media. This influx of information and with the use of a rich computational study as SA provides valuable information to marketers, managers or even manufactures on what consumers or employees perceive or demand (Xu et al., 2019). Thus, opinions derived from the study can change the policy of an organization as well as strategies to meet consumer demands. Nevertheless, SA is limited to deriving the overall polarity other than the reference aspect resulting in the development of SA branches such as targeted sentiment analysis (TSA) (Barnes and Klinger, 2019). Errors that may result from SA are harboured using TSA as the sentiment polarity identification (e.g. negative, neutral, or positive) is attached to a target in their context sentence (Pang and Lee, 2008; Jiang et al., 2011; Saeidi et al., 2016). For example, *"I hate his late coming behaviour, but he is an exceptional performer"* here the reviewer has a negative sentiment toward an employee's attendance behaviour but a positive sentiment on the performance. An aggregated sentiment drawn from the targets can allow the users to understand employee capability. This is achieved by the use of a neural network to automatically learn dimensional representation for targets in the sentence context. In this paper, we examine the effect of various design methods on $NoReC_{fine}$ dataset (Øvrelid et al., 2019) by identifying the target opinion and their associated polarity. With Bidirectional Long Short Term Memory (BiLSTM) as base model, we conduct experiments with other gated mechanisms like Gated Recurrent Unit (GRU) accompanied with deep neural layers.

2 Related Work

Several machine learning methods have been used in TSA and despite its initial design with support vector machine (SVM), there has been an achieved progress by utilizing neural networks to encode base features as continuous and low dimensional vectors. Some studies employed Recursive Neural Network in conducting semantic composition for prediction representation while LSTM implemented in modelling the left and right context of a target and concatenate them for prediction results which were essentially not target specific (Jiang et al., 2011; Socher et al., 2011; Dong et al., 2014; Ma et al., 2018; Li and Lu, 2017; Tang et al., 2016).

The target influence on prediction representations were achieved by an attention mechanism approach leading to bidirectional models that interactively learn attention weights on contexts and targets to separately generate target and context (Wang et al., 2016; Chen et al., 2017). In addition, more refined models are being studied to reduce

[1] https://pages.github.uio.no/arthurd/in5550-exam/ (20/05/2020)

information loss in case of multiple words. For instance, to capture word-level interaction within a target and context, a mix of fine and course-grained attention mechanism was employed which could also depict the same context target interaction (Fan et al., 2018).

Transformer architecture modification introduces a lightweight model in TSA, which adopts attention-based encoders in modelling target words and context differently (Song et al., 2019). For sentences with more than one target word, Zhou et al. (2019), utilizes Convolutional Neural Network (CNN) to model explicit dependency between the opinion words; the architecture obtains significant results on multiple target words in a sentence.

Regardless of the advantages of various designs, some studies show high state of the art results on certain dataset than others with regards to its composition (Chen et al., 2017). In addition, most of these architectures have been done on English datasets except a few with an example of a recent study that used fine-grained sentiment analysis on novel Norwegian dataset *NoReC*$_{fine}$ (Øvrelid et al., 2019). Our study will employ the same dataset to estimate the effect of computational TSA design as used on English dataset on the polarity and target of the data.

3 Data

3.1 NoReC dataset

The *NoReC*$_{fine}$ dataset is composed of a training, validation and testing sub-datasets. The data contains labeled sentiment analysis of professionally authored reviews across a wide variety of domains. The labels follow the Beginning - Inside - Outside (BIO) format, which are used to mark the start and end of a target within a sentence - positive or negative (Øvrelid et al., 2019).

Words	Labels
Men	O
den	B-targ-Negative
er	O
svakere	O
enn	O
”	B-targ-Positive
Sammen	I-targ-Positive
for	I-targ-Positive
livet	I-targ-Positive
”	I-targ-Positive
.	O

Table 1: Example of a sentence in *NoReC*$_{fine}$ dataset.

(a) Words inside negative targets.

(b) Words inside positive targets.

Figure 1: Word cloud from *NoReC*$_{fine}$ dataset, without stop-words. Positive and negative targets share similar words like *filmen* or *albumet*, however some patterns seems to differentiate these sentiments, like *Love* and *Fire* keywords. Note that English words are part of the dataset, like the sentence 351: “På ”California” (There Is No End To Love)” parafraseres finske The Rasmus’ hit ”In The Shadows”, av alle ting, og det lånes generelt raust fra nær pop historie.”

In addition, pre-trained word embeddings on a Norwegian-Bokmaal CoNLL17 corpus were used, obtained using a Word2Vec Continuous Skipgram model [2].

3.2 Distribution

The dataset is composed of a total of 8.259 examples, split into training, validation and testing subdatasets.

	Train	Valid	Test	Total	Avg. Length
Sentences	6145	1184	930	8259	16.8
Targets	4458	832	709	5999	2

Table 2: Number of sentences and targets per datasets.

The labels are not equally distributed over the sentences. In fact, the dataset is imbalanced and 90% of the labels are the label *Outside* (O) of a target.

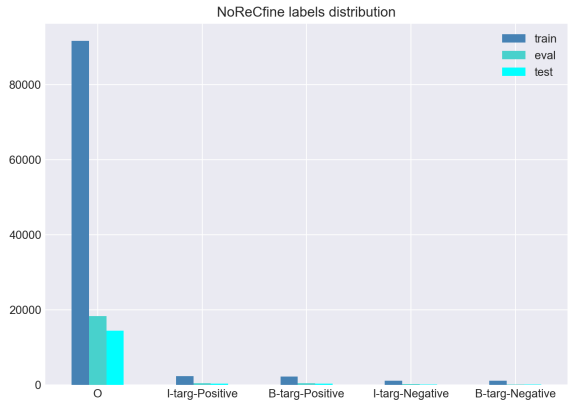


Figure 2: Labels distribution over the train, validation and test *NoReCfine* datasets.

	Train	Valid	Test	Total
O	91691	18336	14425	124452
I-targ-Pos	2339	436	348	3123
B-targ-Pos	2245	433	366	3044
I-targ-Neg	1114	207	117	1438
B-targ-Neg	1094	196	145	1435

Table 3: Imbalanced labels distribution.

Weights can be added to balance the data distribution, and this part will be discussed later.

[2] http://vectors.nlpl.eu/repository/20/58.zip (20/05/2020)

In addition, more than one sentiment can be retrieved from a sentence.

	Train	Valid	Test	Total
O	15.5	15.9	16.1	15.6
I-targ-Pos	3.3	3.0	3.0	3.2
B-targ-Pos	1.4	1.3	1.4	1.4
I-targ-Neg	3.1	3.0	2.3	3.0
B-targ-Neg	1.2	1.1	1.2	1.2

Table 4: Number of labels per sentence (mean).

4 Models

4.1 Approach

Our models are based on a provided baseline model (BiLSTM) for the TSA to analyse the polarity of the target words. As the script given did not conform precisely with (Paszke et al., 2019) standards, and the model was struggling to classify the task words, we further developed a compatible *PyTorch* version that had better results. The BiLSTM model was trained on Norwegian Bokmål word embeddings by utilizing random initialization with a one step forward pass and used cross-entropy loss function on an un-padded index to calculate the loss. In general LSTM models are capable of flexibly capturing the semantic relationship that exists between a target and the context words and avoids gradient vanishing (Tang et al., 2016).

To compare the results, a variant of an LSTM; Gated Recurrent Unit (GRU) model was developed. GRU introduces the combination of the input and forget gates resulting into a simplified gate. While among other changes the model also combines the hidden state to the cell state. As there exist mixed findings between LSTMs and GRU performance, the application of GRU on the $NoReC_{fine}$ data could establish an explainable result. The parameters remained virtually the same in both models.

Then, we wanted to explore the possibilities and results with a BERT model template. We preprocessed the $NoReC_{fine}$ datasets and generated masked arrays, but we faced the same problem of over-fitting.

4.2 BiLSTM

Our baseline model is a LSTM with a word embedding layer. We used the *58.zip* pre-trained word embeddings on Norwegian Bokmål, downloaded from NLPL vectors. We created a skeleton (table 5) that we tuned during the grid-search optimization.

Modules	Dimensions	Parameter
Embedding	(23,574, 100)	-
LSTM	(100, H)	-
Dropout	-	p
Linear	($2*H$, C)	-

Table 5: BiLSTM skeleton used. Note that the bidirectional and recurrent layers parameters were also tested during the grid-search.

4.3 BiGRU

After implementing the BiLSTM model, we quickly implemented the BiGRU base model, as only the core recurrent cell changed. Its skeleton can be described similarly in table 6.

Modules	Dimensions	Parameter
Embedding	(23,574, 100)	-
GRU	(100, H)	-
Dropout	-	p
Linear	($2*H$, C)	-

Table 6: BiGRU skeleton used. Note that the bidirectional and recurrent layers parameters were also tested during the grid-search.

4.4 BERT

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) models are widely used in NLP, from targeted sentiments to question answering. It can work by paired-sentences: from one sentence, it learns to predict the second. However, we did not used this functionality as sentences are independent.

BERT models use a Masked Language Model (MLM) with attention masked arrays on top of the labels (in our case, BIO labels). In addition, special tokens *[CLS]* and *[SEP]* are used to delimit a sentence, a feature mainly used with paired-sentences training.

Modules	Dimensions	Parameter
Word Embeddings	(119,547, 768)	-
Position Embeddings	(512, 768))	-
Token Embeddings	(2, 768)	-
Dropout	-	p
Encoders	[12, 24]	-
Linear	(768, 768)	-
Dropout	-	p
Linear	(768, C)	-

Table 7: BERT skeleton used.

For this short project, we used the $BERT_{BASE}$ (made of 12 encoders) multi-lingual pre-trained model on 104 language (latest model). We also used $BERT_{LARGE}$ (made of 24 encoders) for comparison, but as it is time-consuming to train, we did not experiment with a lot of hyper-parameters tuning.

A special data processing was implemented in *PyTorch*, so we could use pre-trained transformers from *HuggingFace* repository (Wolf et al., 2019).

4.5 Loss Function

We first used a Cross Entropy Loss function, but a majority of models were over-fitting due to the imbalanced data. Thus, we tried a weighted loss which improved the results but did not solve over-fitting issues on baseline models.

The weights were determined from the distribution (table 3).

Target	Weight
O	0.0677
I-targ-Positive	0.9766
B-targ-Positive	0.9772
I-targ-Negative	0.9892
B-targ-Negative	0.9893

Table 8: Weights used for the Cross Entropy Loss function.

5 Results

5.1 Scores

We used normalized confusion matrices (eq. 1) and macro F_1 scores to measure models' errors.

$$\mathcal{N}(M) = (m'_{i,j})_{(i,j)\in[\![0,n-1]\!]^2} \tag{1}$$

$$\textit{where},$$

$$\forall (i,j) \in [\![0,n-1]\!]^2, \quad m'_{i,j} = \frac{m_{i,j}}{\sum_{k=1}^{n-1} m_{k,j}}$$

Normalization makes classes proportional, so that each row of the matrix is summed to 1. When working with imbalanced datasets, normalization enables a better comprehension of the predictions, in terms of percentage. This highlights the mismatched classes, even if they are under-estimated in the data.

5.2 Grid Search

To visualize the impact of a hyper-parameter variation, we fixed all other parameters. By default, we used a learning rate $l_r = 0.1$, a batch size $B_{size} = 64$, a hidden dimension $H = 300$, a bidirectional state $B_{direction} = True$ and $n_{layers} = 2$ layers for the recurrent cell and finally the weights w_L defined in (table 8) for the loss function, trained on 100 epochs. For this section, we implemented an independent package to tune any *PyTorch* model. We only ran the grid-search on four model types, but this package can be used for further projects. The grid-search results are resumed in (Appendix, table 10).

For BiLSTMs and BiGRUs, results are not really satisfying as loss diverges at epoch 20 maximum (Appendix, Figure 4). Even with weights, the model struggles to correctly classify underrepresented targets (*B-targ-Negative* for example). However, as the class *O* is over-represented, global accuracy converges, no matter the parameters combination that was used. The confusion matrices (Appendix, figure 5) highlight these issues - baseline models over-fit the data even with a weighted loss function.

The grid search resulted in a variety of models, and we kept the best ones for each structure (table 9). The BiGRU is our top F_1 score model with the lowest loss.

5.3 Over-fitting

The major issue we faced was over-fitting and over-prediction of *O* labels as highlighted by the confusion matrix in figure 3. Baseline models are not efficient when it comes to classifying imbalanced data, and even though global accuracy converges, the loss (and F_1 score) diverges.

5.4 Error analysis

Figure 5 (Appendix) shows that the most common errors for our different labels are the over-prediction of *Outside (O)* labels. A significant proportion of meaningful labels (*I-targ, B-targ*) are predicted as *O* label.

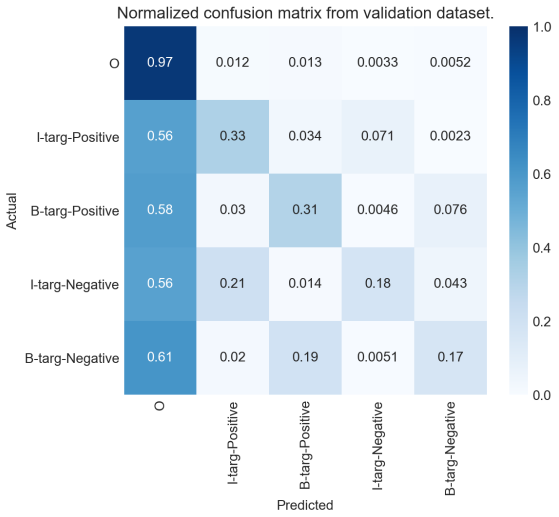


Figure 3: Normalized confusion matrix of the best BiGRU structure, trained over 100 epochs. The label *O* is over-predicted, even with a weighted loss. Meaningful labels like *B-targ, I-targ* are classified in majority as *O*.

On the one hand, our models classify correctly *O* label in more than 96% cases. On the other hand, other classes are classified as *O*.

With non-weighted loss, the over-prediction of *O* drastically increase, as they compose the majority of labels. Attributing weights reduces this phenomenon, but does not solve it fully.

The BiGRU seems to predict the label *O* for the majority of the time. The model seemed to struggle more with correctly predicting negative labels (*I-targ-negative* and *B-targ-negative*) compared to the prediction of positive labels (*I-targ-positive and B-targ-positive*). In addition, our BERT template did not achieve more than a 0.12 F_1 score with a respective 0.97 accuracy.

We did not have enough time to change or try different BERT structures, like RoBERTa with different folds that seems to work well for sentiment extraction - our main issue.

Last but not least, the dataset is more complex to handle as more than one sentiment can be extracted from a sentence. As there is only one *B-targ-Sentiment* label for a sentiment, if the begin-

	Loss	Accuracy	F_1 score	Recall
BiLSTM	0.4582	0.8889	0.2884	0.3527
BiGRU	**0.2415**	**0.9371**	**0.3501**	**0.4351**
BERT-base-multilingual-cased	0.03106	0.9717	0.1157	0.1657
BERT-large-multilingual-cased	0.03320	0.9602	0.0354	0.0152

Table 9: Results of best models from different structures, evaluated on the validation dataset.

ning is mismatched the F_1 score will drop significantly. We tried to add different weights to tackle this issue, with a significant improve (+0.1 F_1 score approximately), but remained under the 0.5 score threshold.

5.5 Conclusion

Targeted Sentiment Analysis is a variant of Sentiment Analysis. In TSA the sentiment towards a specific target in a text is analysed. Starting with a simple baseline model, a grid-search was performed with a BiLSTM and BiGRU to find out optimal parameters. The results showed that the models are suffering with overfitting. The global accuracy remained high (around 0.9), the loss diverged. A possible explanation for this might be that the model predicts the majority class ('label *O*') and therefore correctly classifies a majority, but also miss-classifies a lot of data therefore increasing the loss.

Targeted Sentiment Analysis is a similar topic to Aspect Based Sentiment Analysis, but with a more complex sentiment extraction. Even if opinion mining is a solved research subject, the uncertainty of a starting and ending target makes TSA predictions much more difficult to solve. We figured that simple models (RNN-like) perform better than more complex ones (BERT).

BERT models are well used in Named Entity Recognition areas, and we were convinced that it could outperforms our baseline model. However, because starting tokens depend mainly on the context (and not necessarily on the words) BERT models predict more *O* than reality, leading to a decrease in F_1 score. As also found by Xu et al. (2019), BERT weights may work well with some NLP tasks but fail with review-based tasks.

We did not have enough time to try different tuning of BERT, like a RoBERTa or DistilBERT, which are recently used and may perform better with $NoReC_{fine}$ dataset. For the future we still believe a BERT model might elicit a good performance. Implementing a BERT model is something that could be explored in further research.

Acknowledgements

The models were trained on resources provided by UNINETT Sigma2 - the National Infrastructure for High Performance Computing and Data Storage in Norway. In addition, we used GPU's offered by Google Colab to train and explore other model's architecture.

We would also like to thank the teachers from IFI department at the University of Oslo, who were proactive and supervised both teaching and examination despite the COVID-19 outbreak.

References

Jeremy Barnes and Roman Klinger. 2019. Embedding projection for targeted cross-lingual sentiment: Model comparisons and a real-world study. *CoRR*, abs/1906.10519.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. pages 452–461.

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

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54, Baltimore, Maryland. Association for Computational Linguistics.

Feifan Fan, Yansong Feng, and Dongyan Zhao. 2018. Multi-grained attention network for aspect-level sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3433–3442, Brussels, Belgium. Association for Computational Linguistics.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. pages 151–160.

Hao Li and Wei Lu. 2017. Learning latent sentiment scopes for entity-level sentiment analysis. In *AAAI*.

Dehong Ma, Sujian Li, and Houfeng Wang. 2018. Joint learning for targeted sentiment analysis. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2:1–135.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle,

A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Marzieh Saeidi, Guillaume Bouchard, Maria Liakata, and Sebastian Riedel. 2016. Sentihood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods.

Richard Socher, Cliff Lin, Andrew Ng, and Christoper Manning. 2011. Parsing natural scenes and natural language with recursive neural networks. pages 129–136.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. 2019. Targeted sentiment classification with attentional encoder network. *Lecture Notes in Computer Science*, page 93–103.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective LSTMs for target-dependent sentiment classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3298–3307, Osaka, Japan. The COLING 2016 Organizing Committee.

Junhui Wang, Xiaotong Shen, Yiwen Sun, and Annie Qu. 2016. Classification with unstructured predictors and an application to sentiment analysis. *Journal of the American Statistical Association*, 111(515):1242–1253.

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

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. *BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis*.

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2019. Evaluating commonsense in pre-trained language models.

Lilja Øvrelid, Petter Mæhlum, Jeremy Barnes, and Erik Velldal. 2019. A fine-grained sentiment dataset for norwegian.

Appendix

	Batch sizes			
	32	*64*	*128*	*256*
BiLSTM	**0.3252**	0.3068	0.2988	0.2813
BiGRU	**0.3260**	0.3203	0.3116	0.2963
BERT-base-multilingual-cased	0.1012	**0.1035**	0.0981	0.0516
BERT-large-multilingual-cased	**0.0230**	0.0094	0.0008	0.0000

	Learning rates			
	0.01	*0.05*	*0.1*	*0.2*
BiLSTM	0.3158	**0.3173**	0.2992	0.2991
BiGRU	0.2988	0.3098	**0.3255**	0.3130
BERT-base-multilingual-cased	0.0878	0.0965	**0.1129**	0.1013
BERT-large-multilingual-cased	0.0075	**0.0103**	0.0056	0.0012

	Dropouts			
	0.1	*0.2*	*0.3*	*0.4*
BiLSTM	0.3059	0.2964	0.3101	**0.3264**
BiGRU	0.3069	0.3170	0.3255	**0.3422**
BERT-base-multilingual-cased	0.1095	0.0899	0.1053	**0.1136**
BERT-large-multilingual-cased	0.0102	**0.0124**	0.1053	0.0114

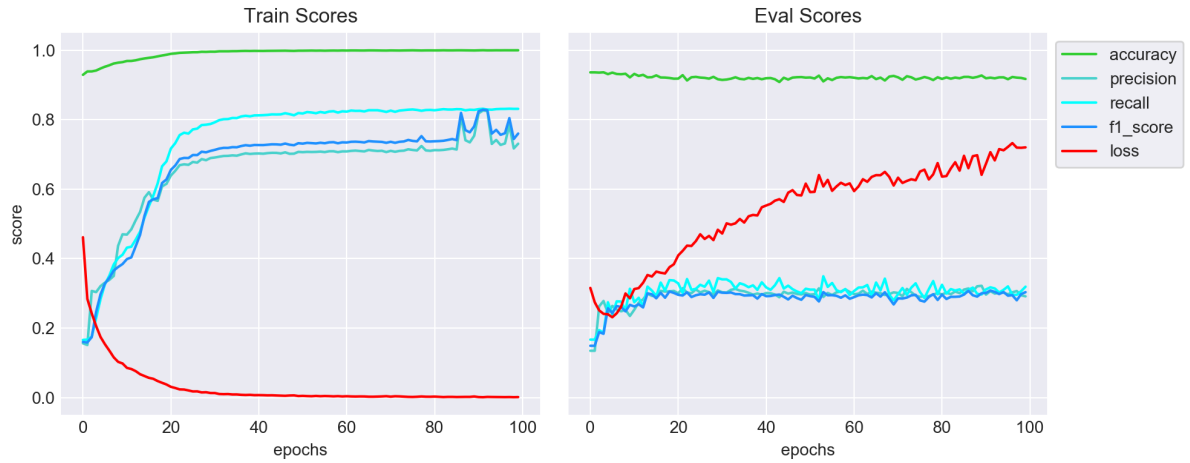
	Weighted Loss Function	
	No weights	*Weights*
BiLSTM	0.3175	**0.3183**
BiGRU	0.3307	**0.3510**
BERT-base-multilingual-cased	**0.1103**	0.0156
BERT-large-multilingual-cased	0.0026	**0.0051**

	Hidden dimensions			
	100	*200*	*300*	*400*
BiLSTM	0.2962	**0.3210**	0.3074	0.3191
BiGRU	0.2775	**0.3165**	0.3071	0.3059

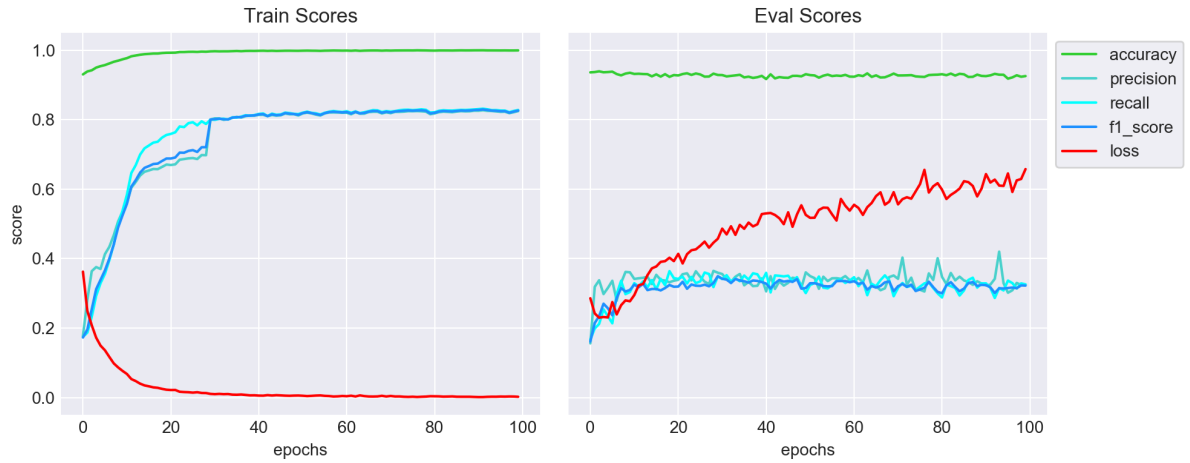
	Bidirectional	
	True	*False*
BiLSTM	**0.3183**	0.2698
BiGRU	**0.3173**	0.2577

	Number of layers			
	1	*2*	*3*	*4*
BiLSTM	0.3002	0.3031	**0.3066**	0.2986
BiGRU	**0.3206**	0.3162	0.3161	0.3109

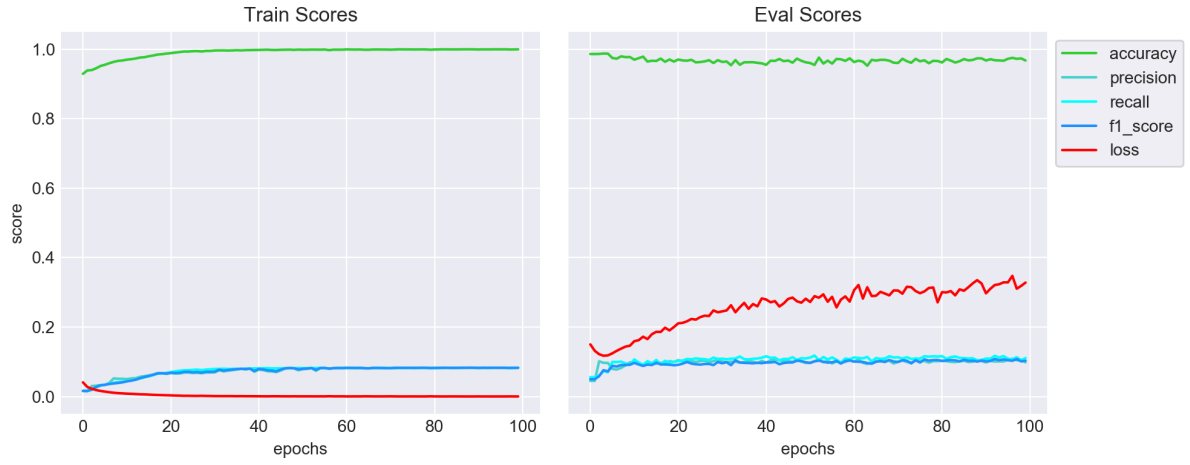
Table 10: Results of the grid-search. The table shows F_1 score regarding a variation of one hyper-parameter. All scores were obtained on the validation dataset.



(a) BiLSTM results.

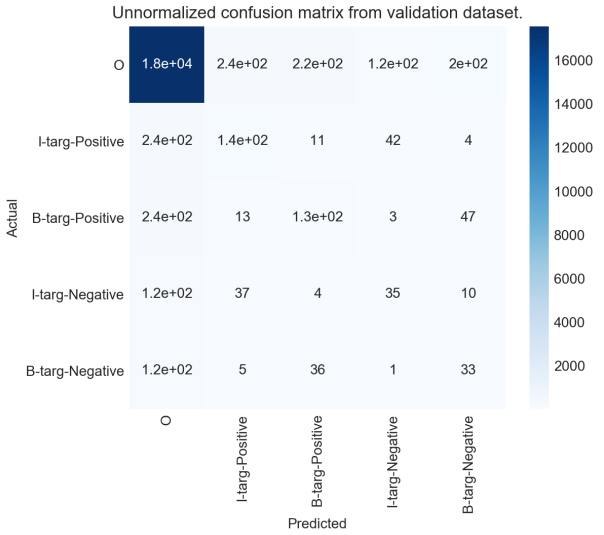


(b) BiGRU results.

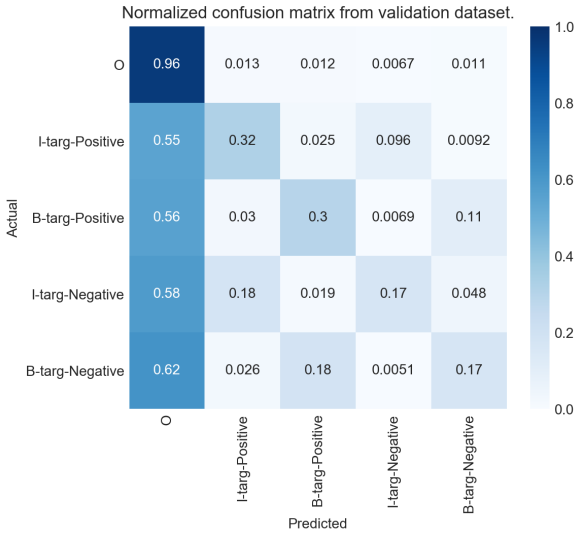


(c) BERT-base-multilingual results.

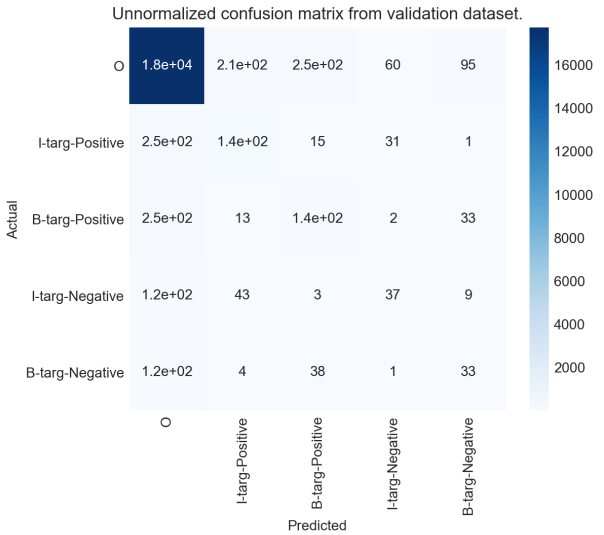
Figure 4: Results of best models from different structures, evaluated on the validation dataset. Even though the accuracy converge in all cases, loss increases when models are training for too long. The shift occurs relatively soon (5-20 epochs), and we experienced that it happens regardless of our hyper-parameters tuning.



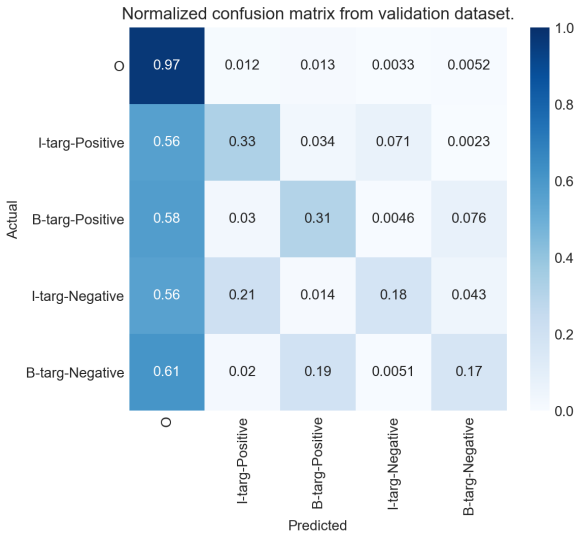
(a) Unnormalized BiLSTM confusion matrix.



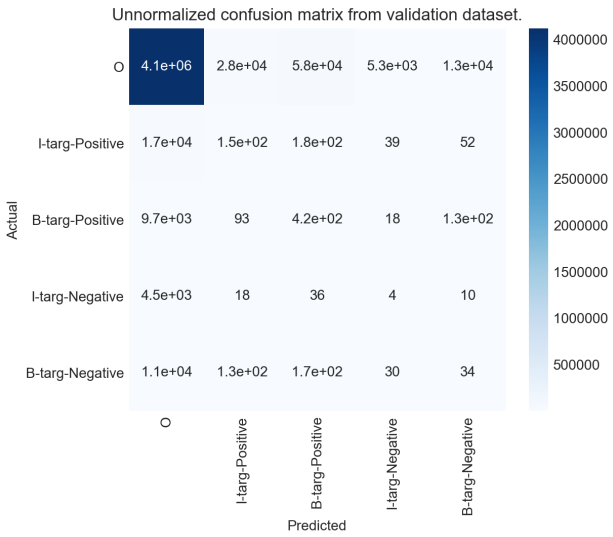
(b) Normalized BiLSTM confusion matrix.



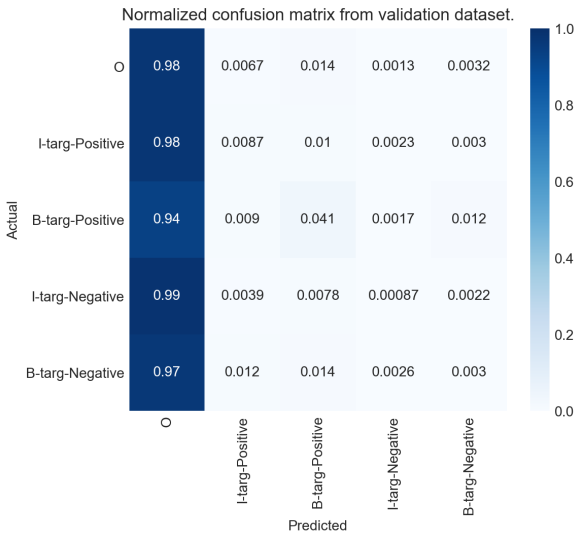
(c) Unnormalized BiGRU confusion matrix.



(d) Normalized BiGRU confusion matrix.



(e) Unnormalized BERT confusion matrix.



(f) Normalized BERT confusion matrix.

Figure 5: Unnormalized (left) and normalized (right) confusion matrices of the best model taken from each structures, evaluated on the validation dataset.