Evaluating Unsupervised Representation Learning for Detecting Stances of Fake News

Maike Guderlei and Matthias Aßenmacher

Department of Statistics, Ludwig-Maximilians-Universität, Munich, Germany

Data

For our experiments, we used data from the *Fake News Challenge Stage 1* (FNC-1) (accessible via `http://www.fakenewschallenge.org/`), which treats Fake News detection as a classification task with four categories. More specifically, FNC-1 is conceptualized as a stance detection task with the claim being treated as a headline and the stance of the news article body being either *Agree*, *Disagree*, *Discuss* or *Unrelated*. It is thus an important pre-step in identifying Fake News and exploring how AI tools can be leveraged in combatting them. If most news articles agree with a claim, this can be interpreted as an indicator of the truthfulness of the claim. On the contrary, if a lot of news disagree with the claim, the claim is likely Fake News. FNC-1 ARC extends the FNC-1 data set by adding data from user posts, so that it eventually comprises $\sim$ 64k instances (vs. FNC-1 only $\sim$ 50k).

Models

Recent state-of-the-art NLP models, namely BERT [1], RoBERTa [2], ALBERT [3], DistilBERT [4] and XLNet [5], were evaluated with respect to their ability of successfully performing this task. In doing so, we put a special focus on exploring different freezing techniques as well as on finding promising configurations for a chosen set of hyperparameters.

Objectives

- **Initial Experiments:** Analyze the suitability of different freezing techniques for finetuning
- **Detailed Grid Search:**
  - Understand hyperparameter tuning for finetuning in the context of Fake News
  - Compare autoencoder (AE) to autoregressive (AR) models

Initial Experiments

- Fix sequence length (512), batch size (8), learning rate (3e-5) and learning rate schedule (linear)

Exploration of different freezing techniques:

1. `Freeze`: finetune only last projection- & classification-layer
2. `No Freeze`: finetune whole model
3. `Freeze Embed`: finetune whole model, *except* for the embedding layers

	Freeze	No Freeze	Freeze Embed
BERT	20.88	**75.62**	74.93
RoBERTa	20.88	79.27	**81.72**
DistilBERT	20.88	**76.57**	76.46
ALBERT	34.66	67.91	**68.16**
XLNet	27.51	80.95	**82.18**

Table 1:Mean macro-avgd. F_1 (F_1-m) over three runs. Results on the dev set of a train/dev split of the FNC-1 training set. FNC-1 ARC results omitted due to lacking space, but similar.

Detailed Grid search

Hyperparameter	Considered Configurations
Batch size/Sequence length	16/256; 32/256; 4/512; 8/512
Learning rate	1e-05; 2e-05; 3e-05; 4e-05
Learning rate schedule	constant, linear, cosine

Table 2:Search space over chosen hyperparameters. Sequence length and batch size depend on one another due to memory capacity reasons. For the longer sequence length only smaller batch sizes could be considered. All learning rate schedules use a warmup period of 6% of the total optimization steps.

Results

- `Freeze Embed` best combination of performance and finetuning time
- RoBERTa (AE) beats XLNet (AR)
- Learning rate most important hyperparameter
- Longer sequence length often prefers a higher batch size except for smaller learning rates

Results of the detailed Grid Search

		BERT		RoBERTa		DistilBERT		ALBERT		XLNet	
	LR	Winner	F_1-m	Winner	F_1-m	Winner	F_1-m	Winner	F_1-m	Winner	F_1-m
FNC-1	1e-5	16,256,cos	62.46	4,512,lin	**78.18**	8,512,cst	65.72	4,512,lin	56.62	4,512,cos	73.47
	2e-5	16,256,cst	**70.18**	16,256,lin	76.54	16,256,lin	67.64	8,512,cos	59.74	16,256,cos	**75.00**
	3e-5	16,256,cst	69.36	32,256,cos	76.52	32,256,cst	69.64	16,256,lin	**59.80**	32,256,cos	73.27
	4e-5	8,512,linear	68.09	32,256,lin	74.84	32,256,cst	**72.11**	16,256,lin	58.33	32,256,lin	73.46
FNC-1 ARC	1e-5	8,512,lin	68.87	4,512,lin	**78.19**	8,512,lin	71.99	8,512,cst	63.40	4,512,linear	74.42
	2e-5	4,512,lin	**72.20**	8,512,lin	77.27	8,512,cst	**73.59**	8,512,cos	**65.01**	8,512,lin	75.47
	3e-5	8,512,cos	70.93	16,256,lin	77.54	32,256,lin	72.99	16,256,lin	64.67	16,256,lin	73.97
	4e-5	32,256,lin	70.83	32,256,lin	77.54	16,256,lin	73.13	32,256,lin	63.63	32,256,lin	**75.57**

Table 3:Winning configuration (chosen with respect to F_1-m on the evaluation set) out of the 12 possible configurations per LR. *Winner* columns show batch size, sequence length and LR schedule (in this order). F_1-m of the winning configuration per model indicated in bold, values in teal indicate the overall winning configuration per data set. The overall winning configuration over both data sets is additionally marked by a box. All reported values are obtained on the official test set.

Conclusion

Even with minimal hyperparameter tuning and only finetuning for 3 epochs, the models already performed considerably well on both data sets. It is important to *not only* finetune the classification and pooling layers that are stacked on top of the pretrained models. The most important hyperparameter is the learning rate. Furthermore, the models are relatively robust with respect to the learning rate schedule, the batch size, as long as it is adjusted to the learning rate and to a certain degree also the sequence length. The excessive pretraining approach of RoBERTa can outperform the permutation language model objective of XLNet, which may be due to the segment-level nature of the task.

References (Poster)

[1] J. Devlin *et al.*, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *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)*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[2] Y. Liu *et al.*, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[3] Z. Lan *et al.*, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.

[4] V. Sanh *et al.*, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[5] Z. Yang *et al.*, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in neural information processing systems*, pp. 5753–5763, 2019.

Further Information // Contact

- Web:
  Working group: misoda.statistik.uni-muenchen.de
  Code: github.com/magud/fake-news-detection
- E-Mail:
  M.Guderlei: maike@guderlei.de
  M. Aßenmacher: matthias@stat.uni-muenchen.de