TNT: Text Normalization based Pre-training of Transformers for Content Moderation

Fei Tan, Yifan Hu, Changwei Hu, Keqian Li, Kevin Yen
Yahoo Research, New York, NY, USA
{fei.tan, yifanhu, changweih, keqian.li, kevinyen}@verizonmedia.com

Abstract

In this work, we present a new language pre-training model TNT (Text Normalization based pre-training of Transformers) for content moderation. Inspired by the masking strategy and text normalization, TNT is developed to learn language representation by training transformers to reconstruct text from four operation types typically seen in text manipulation: substitution, transposition, deletion, and insertion. Furthermore, the normalization involves the prediction of both operation types and token labels, enabling TNT to learn from more challenging tasks than the standard task of masked word recovery. As a result, the experiments demonstrate that TNT outperforms strong baselines on the hate speech classification task. Additional text normalization experiments and case studies show that TNT is a new potential approach to misspelling correction.

1 Introduction

Language model pre-training (self-supervised or unsupervised learning) has been a popular thread in Natural Language Understanding (NLP) studies recently due to its universal representation capacity (Radford et al., 2018; Devlin et al., 2019; Radford et al., 2019; Brown et al., 2020). It has been thus widely used in a multitude of language processing tasks such as named entity recognition, sentiment analysis, question answering and content moderation (Bodapati et al., 2019). In addition, the masking pre-training paradigm introduced by BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) has been employed for other tasks such as image processing (Trinh et al., 2019), optical flow (Liu et al., 2019a), and audio-visual co-segmentation (Rouditchenko et al., 2019).

Recently, many variants have been proposed to further improve the pre-training procedure (Liu et al., 2019b; Wang et al., 2019; Sun et al., 2019). They have also advanced the state-of-the-art performance on multiple downstream natural language understanding tasks (Leaderboard) consistently. Almost all these studies train language models by predicting the masked words in different manners. The underlying mechanism is *Cloze task* (Taylor, 1953). The pre-training model itself, however, has not been fully exploited to address complicated yet feasible tasks. It is reasonable to expect that models can learn a better universal language representation if the pre-training procedure can be aligned with more challenging tasks.

In this article, we attempt to improve the language representation by proposing TNT: **T**ext **N**ormalization based pre-training of **T**ransformers. TNT enhances the language learning by utilizing text normalization pre-training objective, inspired by misspelling correction. Specifically, TNT randomly manipulates tokens from the input text. The objective is then to reconstruct the original tokens of the manipulated words based on the context by predicting both recovery operation type and original token labels as illustrated in Fig. 1. Unlike the masked language model, TNT has to offer two predictions to reconstruct the original text. In addition, TNT does not have to be given the prediction positions in advance. This aligns with the fact that misspelling correction needs to perform the position-agnostic prediction for both aspects.

Perpetrators often intentionally obfuscate certain words about groups, or abusive words, by misspelling, or leetspeak (e.g.,"f@ggot","ph*ck", "w.e.t.b.a.c.k.") (Perea et al., 2008). This could sidestep the content moderation algorithms easily as exemplified in Table 5. To assess the learning capacity of TNT for obfuscated text and reduce the training cost, it is pre-trained only on one dataset and then applied to three datasets related to a hate speech detection task. TNT achieves better results

compared to strong baselines on these datasets. Furthermore, we conduct an experiment on misspelling correction, and demonstrates that TNT has appealing language understanding capacity.

Our contributions are summarized as follows: (1) we introduce text normalization into the language training paradigm, which involves challenging tasks of predicting both operation types and token labels; (2) we show that TNT advances the challenging downstream text classification task, which benefits the content moderation; (3) TNT offers a new perspective on misspelling correction.

2 Related Works

BERT (Devlin et al., 2019) is developed by introducing the bidirectional encoder of well-known transformers to learn the contextual representation of text, which is underpinned by the attention mechanism (Vaswani et al., 2017). It randomly masks a certain portion of tokens from the input and then learns to predict these masked words. This cloze task based pre-training strategy enables BERT to advance the state-of-the-art performance on various key NLP tasks. It also inspires the community with a plethora of subsequent works (Yang et al., 2019; Sun et al., 2019; Wang et al., 2019; Liu et al., 2019b). Among them several are closely related to our approach: XLNet (Yang et al., 2019) and StructBERT (Wang et al., 2019) improve the masking by imposing the permutation and shuffling among words and sentences. StructBERT is one of the current state-of-the-art algorithms topping the GLUE leaderboard[1] (Leaderboard), and is most similar to this work. TNT differs from StructBERT in that it is inspired by the need for misspelling correction, and therefore not only allows permutation of words, but also deletion and insertion.

For online abusive language moderation, BERT has also been shown effective and advances the overall performance largely (Bodapati et al., 2019). In addition, early works formulate the hate speech detection as the generic text classification, alternatively focus on certain ethnic groups or building up blacklists of swear words (Nobata et al., 2016a; Badjatiya et al., 2017). Misspelling correction is also a long-standing problem in NLP (Hirst and Budanitsky, 2005; Bassil, 2012; Islam and Inkpen, 2009) and has been widely used in real-word scenarios like word processing system and email spell checking.

Table 1: Real user manipulated text examples. 0, 1, 2 and 3 in the bracket correspond to recovery operation types: substitution, transposition, deletion and insertion. Identity (4) is also exemplified here for completeness

Manipulation	Text	Normalization
substitution	I damn sure didn't vote for the Marxist 8astard!	$8 \rightarrow b(0)$
transposition	mario= dumb cutn	$\text{t} \leftrightarrow n(1)$
deletion	Don Lemon is a shthead.	i (3)
insertion	Pi–ss on Putin !!!!!!!!!!___!!!!!!!!!!	$\not{}(2)$
identity	She is a nice lady.	c(4)

3 Pre-training

3.1 Architecture

We develop TNT based on a multi-layer bidirectional Transformer network as encoder used in popular language models like BERT (Devlin et al., 2019). The well-known masking strategy employed in BERT and subsequent works is inspired by cloze task for human-level language understanding. People are required to fill out the omitted words from a passage based on context.

Unlike the cloze procedure, text normalization involves more diversified and challenging tasks. The goal is to rewrite a sentence that was not properly formed, either due to misspelling, or due to intentional manipulation. A perpetrator would misspell on purpose, with the intention of evade detection, through *substitution*, *transposition*, *deletion* and *insertion* as exemplified in Table 1. The task of text normalization is to understand the manipulated sentence, and normalize it to the correct form. Therefore in TNT, motivated by the task of text normalization, we propose the pre-training tasks of substitution, transposition, deletion and insertion. The masking procedure could be viewed as a special case of the text normalization objective, under substitution.

For substitution (e.g., masking) and insertion, the corresponding *token label* predictions alongside *operation type* is required to reconstruct text. Specifically, we have normalization type $o \in \mathcal{O} = \{0, 1, 2, 3, 4\}$, where o is the operation type with values 0 (substitution), 1 (transposition), 2 (deletion), 3 (deletion) or 4 (identity), as exemplified in Table 1. The normalized token labels $l \in \mathcal{V}$

[1]The General Language Understanding Evaluation (GLUE) benchmark is a collection of resources for training, evaluating, and analyzing natural language understanding systems. It consists of 9 sentence- or sentence-pair language understanding tasks.

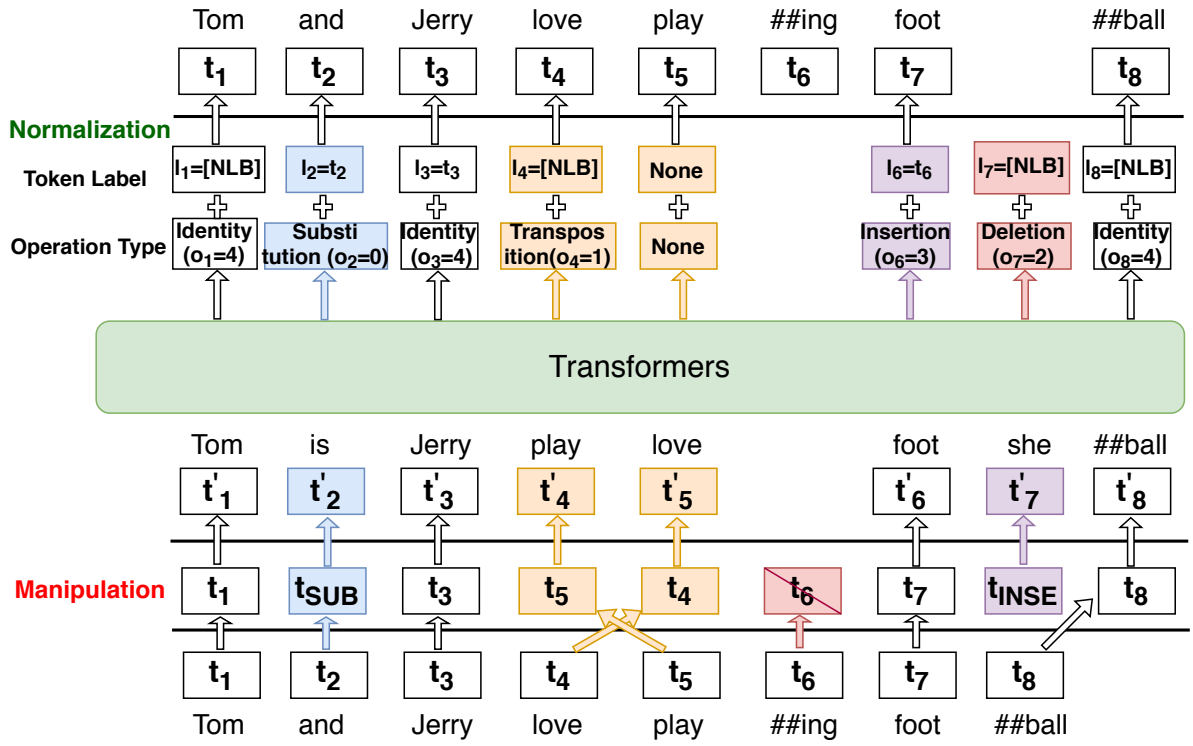

Figure 1: Wordpiece illustration of two pre-training subtasks in TNT. "None" means no prediction as the preceding transposition covers it already. [CLS], [SEP], positional and segment embedding are omitted for brevity.

are ground-truth tokens for corresponding types, where $\mathcal{V}$ is the vocabulary. It is noted that for transposition, deletion and identity, no token labels are required to reconstruct the original text. Thus, we introduce a special token symbol [NLB] as the placeholder. Fig. 1 illustrates the generation procedure of pre-training instances and the joint training of two subtasks.

3.2 Objective

Given an input sequence with manipulated tokens, the operation type and token label objectives can be denoted simply as $\mathcal{O}_{operation} = \arg\max_{\boldsymbol{\theta_o}} \sum_{i=1}^{M} w_i^o \log \mathrm{P}(o_i|t_i'; \boldsymbol{\theta_o})$ and $\mathcal{O}_{label} = \arg\max_{\boldsymbol{\theta_l}} \sum_{i=1}^{M} w_i^l \log \mathrm{P}(l_i|t_i'; \boldsymbol{\theta_l})$, respectively. t_i' is the observed token from the input sentence. o_i and l_i are ground-truth operation type and token label as illustrated in Fig. 1. M is the maximum length of input sequence. $\boldsymbol{\theta_o}$ and $\boldsymbol{\theta_l}$ are sets of trainable parameters. w_i^o and w_i^l are weights for operation type and token label, respectively. The overall pre-training objective is $\mathcal{O} = \mathcal{O}_{operation} + \mathcal{O}_{label}$.

4 Experiments

We perform the hate speech classification and misspelling correction tasks based on the pre-trained model.

4.1 Datasets

Our primary dataset is extracted from user comments on Yahoo News and Finance, and consisted of 1.43M labeled comments. Among them, 7% of the comments are labelled as abusive (including hatespeech and profanity). The labeled data were collected as follows: comments that are reported as "abusive" for any reason by users of Yahoo properties are sent to in-house trained raters for review, and the decisions of the raters form the labels. Further details can be found at (Nobata et al., 2016b). In addition, we experimented on two publicly available hatespeech datasets: Twitter, and Wikipedia (Wiki) (Agrawal and Awekar, 2018). Wiki set here is a collection of discussions among editors on talk pages for improving Wiki articles, which includes inflammatory posts. The statistics of three datasets are shown in Table 2.

We split the dataset into train/development/test sets with a ratio 70%/10%/20%. We generate vocabulary, pretrain the language modeling tasks, and train the hate speech prediction task using only the training set. We tune hyper-parameters on the development set, and report final results on the test set.

Table 2: Basic statistics of datasets

Source	# Abusive	# Clean	Total	% Abusive
Yahoo	100,652	1,328,486	1,429,138	7.04%
Twitter	5,054	11,036	16,090	31.4%
Wiki	13,590	102,274	115,864	11.7%

4.2 Experiment Setups

For TNT pre-training, the substitution follows the masking setting in BERT. We also set up 5% manipulation rate for transposition, deletion and insertion, respectively. The rest of the tokens remain unchanged.

The vocabulary generation, wordpiece tokenization, learning rate, weight decay, warm-up and other training settings follow BERT. The model size is reduced to quarter of the original BERT. The maximum length of input sequence M is set to 256. Parameter scale is $O = (V + M + S) \times H + L \times 12H^2 + H^2$ where V, S are vocabulary size $|\mathcal{V}|$, segment type size. H and L are the hidden layer dimension and the number of transformer block layers, respectively.

We mainly report two models with wordpiece and character inputs as detailed in Table 3. The main difference is that manipulation and normalization are performed on different levels. Wordpiece- and character-level operations are exemplified in Fig. 1 and Table 1, respectively.

Table 3: Parameter scale and weight settings, character for misspelling correction. Tuples of w_i^o and w_i^l are weights associated with operation type {0,1,2,3,4}

Model	V	F	S	L	H	O	w_i^o	w_i^l
Wordpiece	40K	256	1	3	192	9.09M	(1,1,1,1,0)	(1,0,0,1,0)
Character	6.8K	512	1	3	192	2.77M	(1,1,1,1,1)	(1,0,0,1,0)

For both TNT models, we run the pre-training procedure for 64 epochs. We pre-train quarter size of BERT (wordpiece) and StructBERT (wordpiece) with the same dataset, epochs and wordpiece vocabulary. In the fine-tuning phase, batch size, the number of batches and learning rate are set to 64, 10 and 2e-5. For all models, we discard the sentence-pair training objective.

4.3 Results

For the downstream hate speech classification, we do fine-tuning on top of aggregate embedding [CLS] of wordpiece TNT as BERT does. Threshold-free AUC@ROC and AUC@PR[2] (Davis and Goadrich, 2006) and threshold (0.5 used here) based F1 score and Matthews correlation coefficient (MCC) are adopted. MCC and AUC@PR are generally regarded as balanced metrics.

[2]Precision-Recall

Table 4: Performance comparisons on wordpiece TNT for hate speech classification

Source	Method	AUC@ROC	AUC@PR	MCC	F1 Score
Yahoo	BERT	93.92	69.56	60.34	61.10
	StructBERT	94.04	70.22	60.66	61.15
	TNT	94.30 (↑0.26)	70.91 (↑0.69)	62.04 (↑1.38)	63.18 (↑2.03)
Twitter	BERT	89.52	81.10	62.38	73.99
	StructBERT	89.96	82.77	62.58	74.09
	TNT	90.42 (↑0.46)	82.81 (↑0.04)	63.99 (↑1.41)	75.29 (↑1.20)
Wiki	BERT	97.10	87.07	76.04	78.67
	StructBERT	97.10	87.07	75.59	78.37
	TNT	97.30 (↑0.20)	87.86 (↑0.79)	76.32 (↑0.73)	79.10 (↑0.73)

Table 5: Case studies for different models

Case	BERT	StructBERT	TNT	Text
1	0.0434	0.0653	0.7401	If njjjgers are killing more people daily..why in the world would anyone want gun control.
2	0.421	0.9275	0.9651	Buyout Announced - $45/share massfaggot
3	0.2445	0.4729	0.8083	Good, we don't want a foggat President anyway
4	0.0925	0.0902	0.622	Dey BigFatAceNeggirBeach be's LOUD!!! lol!!!

We first report the comparison results on Yahoo set. As shown in Table 4, our TNT outperforms BERT and StructBERT on test set. Particularly, it achieves more than 1% and 2% improvement in terms of MCC and F1 score, respectively. The performance gain could be reasonably attributed to the new training objective. We further do fine-tuning based on pre-trained models directly for Twitter and Wikipedia sets, respectively. The advantage of TNT over baselines still holds. The superiority of TNT on classification task over multiple datasets signalizes that text normalization based training strategy is a promising direction for better universal language representation learning. Although all three sets are user generated content, Wiki users are somewhat different. As Wiki itself is a collaborative knowledge repository, editors are likely to attack others due to disputes on specific domain knowledge. However, the users are the general public who post comments and tweets more casually for Yahoo and Twitter. In this context, text misspellings in Wiki are likely to be less severe and intentional than others. The way we develop the model enables it to learn better representations especially for garbled text compared to standard masking schemes. Thus, the performance gain is more salient in Yahoo and Twitter.

To better understand their performance difference intuitively, we illustrate in Table 5 some specific error case analysis, where toxic comments are created by users to attack a certain group of people. The key parts are all intentionally manipulated to

Table 6: Misspelling correction comparison examples. For Bing Spell Check, we render the best results from Proof and Spell modes. The corrected results of Google Docs and Grammarly are based on their top suggestions

Case	Text	Autocorrect	Bing Spell Check	Google Docs	Grammarly	TNT
1	M ke Y*h0! Graet Aga 1n	M he Y*h0! Great Age 1n	M ke Y*h0! Graet Aga 1n	M ke Y*h0! Great Aga 1n	M ke Y*h0! Great Aga 1n	Make Yahoo Great Again
2	T Ⓡump anf B*!den	T Ⓡump and B*!den	Trump and Biden	T Ⓡump and B*!den	T Ⓡump and B*!den	Trump and Biden
3	UAS is a great county	Was is a great county	USA is a great county	UAS is a great county	UAS is a great county	USA is a great country
4	UAS stands for Unmanned Aircraft Systems	Was stands for Unmanned Aircraft Systems	UAS stands for Unmanned Aircraft Systems	UAS stands for Unmanned Aircraft Systems	UAS stands for Unmanned Aircraft Systems	UAS stands for Unmanned Aircraft Systems
5	she recieved her prize	she received her prize	she received her prize	she received her prize	she received her prize	she received her prize
6	we had heard from you more definately	we had heard from you more definitely	we had heard from you more definitely	we had heard from you more definitely	we had heard from you more definitely	we had heard from you more definately

obfuscate moderation algorithms. For case 1 with substitution, TNT functions well but others work poorly. Both StructBERT and TNT work better than BERT for case 2 involving white space deletion. Case 3 comes out with a subtle transposition, TNT performs robust than the other two. Regarding the most challenging case 4 with combination of white space deletion, transposition and substitution, only TNT still works well overall.

5 Misspelling Correction

Misspelling correction is a long-standing research topic (Islam and Inkpen, 2009; Whitelaw et al., 2009; Bassil, 2012) and has been widely commercialized as a service such as Bing Spell Check (Bing) and Grammarly (Grammarly). TNT can be readily employed in misspelling correction. We here evaluate TNT using its character-level[3] variant without additional fine-tuning.

We aggressively misspell the test set of Yahoo in Table 2 by 15% for each sample. Then we employ the pre-trained TNT model to recover the text. As a comparison, we also examine an open-source[4] tool autocorrector (Autocorrect) for reference. Edit distance (Distance), and BLEU (BLEU) are adopted to measure the distance and similarity between corrected samples and original ones as detailed in Table 7. TNT performs significantly better than the dictionary look-up algorithm.

Table 7: Misspelling correction comparison

Metric	Misspelling	Autocorrect	TNT
Edit Distance (↓)	16.9391	16.0261	**4.0115**
Normalized Edit Distance (↓)	0.1295	0.1259	**0.0307**
BLEU (↑)	0.3818	0.4609	**0.8309**

In addition, we cross-check the results between TNT and popular commercial spell check products through case studies as reported in Table 6. Among all tools, Bing leads the performance, followed by Google Docs and Grammarly, and Autocorrect performs the worst. Overall, TNT functions very well particularly for case 1 as a combination of multiple challenging misspellings. It is noted that *received* and *definitely* are two of most commonly misspelled words (Words), but TNT fails on the correction of "definately". Overall, "definitely" is not a strongly contextual word derived from the whole sentence here. The limited training set might restrict the correction capacity as well.

6 Discussions and future work

We conducted experiments on three classification tasks. The data size for Yahoo Finance and News, while being one of the largest in the context of hate-speech classification, is nevertheless small in the context of language modeling. We plan to perform large-scale pre-training and evaluation on GLUE datasets for the comprehensive analysis.

This work targets sentence level language understanding. As far as we know, no data available for misspelled words in the context of sentences, we thus have to generate the evaluation set by ourselves. The main goal here is not to develop a more powerful misspelling corrector, but rather to propose a new and stronger language modeling approach. We thus don't set up the strict and comprehensive evaluation for apples-to-apples comparison on spelling correction. We will continue to explore this line in the future.

7 Conclusion

In this work, we propose a new language representation training strategy TNT. TNT improves language modeling by training a transformer to reconstruct text from four operation types typically seen in text manipulation. We show that when fine-tuned for the content moderation task of detecting hatespeech, the new model performed better than the state of the art baselines. We also demonstrate its effectiveness in misspelling correction.

[3] As misspelling usually involves with many subtle changes, we resort to character sets as the vocabulary with much flexibility.

[4] only free API with large-scale calls

References

Sweta Agrawal and Amit Awekar. 2018. Deep learning for detecting cyberbullying across multiple social media platforms. In *European Conference on Information Retrieval*, pages 141–153. Springer.

Autocorrect. https://github.com/fsondej/autocorrect.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760.

Youssef Bassil. 2012. Parallel spell-checking algorithm based on yahoo! n-grams dataset. *arXiv preprint arXiv:1204.0184*.

Bing. https://azure.microsoft.com/en-in/services/cognitive-services/spell-check/.

BLEU. https://en.wikipedia.org/wiki/BLEU.

Sravan Babu Bodapati, Spandana Gella, Kasturi Bhattacharjee, and Yaser Al-Onaizan. 2019. Neural word decomposition models for abusive language detection. *arXiv preprint arXiv:1910.01043*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Jesse Davis and Mark Goadrich. 2006. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186.

Edit Distance. https://en.wikipedia.org/wiki/Edit_distance.

Grammarly. https://app.grammarly.com/.

Graeme Hirst and Alexander Budanitsky. 2005. Correcting real-word spelling errors by restoring lexical cohesion. *Natural Language Engineering*, 11(1):87–111.

Aminul Islam and Diana Inkpen. 2009. Real-word spelling correction using google web 1tn-gram data set. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1689–1692.

GLUE Leaderboard. https://gluebenchmark.com/leaderboard/.

Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. 2019a. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4571–4580.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016a. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016b. Abusive language detection in online user content. In *WWW*.

Manuel Perea, Jon Andoni Duñabeitia, and Manuel Carreiras. 2008. R34d1ng w0rd5 w1th numb3r5. *Journal of experimental psychology. Human perception and performance*, 34:237–41.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Andrew Rouditchenko, Hang Zhao, Chuang Gan, Josh McDermott, and Antonio Torralba. 2019. Self-supervised audio-visual co-segmentation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2357–2361. IEEE.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Trieu H Trinh, Minh-Thang Luong, and Quoc V Le. 2019. Selfie: Self-supervised pretraining for image embedding. *arXiv preprint arXiv:1906.02940*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Liwei Peng, and Luo Si. 2019. Structbert: Incorporating language structures into pre-training for deep language understanding. *arXiv preprint arXiv:1908.04577*.

Casey Whitelaw, Ben Hutchinson, Grace Y Chung, and Gerard Ellis. 2009. Using the web for language independent spellchecking and autocorrection. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 890–899. Association for Computational Linguistics.

Misspelled Words. https://www.msn.com/en-us/lifestyle/lifestyle-buzz/the-10-most-commonly-misspelled-words-in-the-english-language/ar-BBOS1ZB.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.