Applying SoftTriple Loss for Supervised Language Model Fine Tuning

Anonymous ACL submission

Abstract

We introduce a new loss function TripleEntropy to improve classification performance for fine-tuning general knowledge pre-trained language models based on cross-entropy and SoftTriple loss. This loss function can improve the robust RoBERTa baseline model fine-tuned with cross-entropy loss by about (0.02% - 2.29%). Thorough tests on popular datasets indicate a steady gain. The fewer samples in the training dataset, the higher gain – thus, for small-sized dataset it is 0.78%, for medium-sized – 0.86% for large – 0.20% and for extra-large 0.04%.

1 Introduction

Natural language processing (NLP) is a rapidly growing area of machine learning with applications wherever a computer needs to operate on a text that involves capturing its semantics. It may include text classification, translation, text summarization, question answering, dialogues. All these tasks are upstream and depend on the quality of the text representation (White et al., 2015). Many models can produce such text representations, from Bag-Of-Word or Word2Vec word embedding to the state-of-the-art language representation model BERT with variations in most NLP tasks.

The best performance on text classification tasks is obtained when the model is first trained on a general knowledge corpus to capture semantic relationships between words and then fine-tuned with an additional dense layer on a domain corpus with cross-entropy loss (Radford et al., 2019).

We introduce a new loss function TripleEntropy to improve classification performance for fine-tuning general knowledge pre-trained language models based on cross-entropy loss and SoftTriple loss (Devlin et al., 2018; Qian et al., 2019). Triplet Loss transforms the embedding space so that vector representations from the same class can form separable subspaces, stabilizing, and generalizing the language model fine-tuning process. TripleEntropy can improve the fine-tuning process of the RoBERTa based models so the performance on downstream task increases by about (0.02% - 2.29%).

In the following sections, we review relevant work on state-of-the-art in distance metric learning (Section 2); describe our approach for training and our metric SoftTriple loss and outline the experimental setup (Section 3); discuss the results (Section 4); conclude and offer directions for further research (Section 5).

2 Related Work

2.1 Building Sentence Embeddings

Building embeddings that represent sentences is challenging because the natural language can be very diverse. The meaning can change drastically depending on the context of a word. It is also an important issue because the quality of sentence embeddings substantially impacts the performance of all downstream tasks like text classification and question answering. Because of that, so far, considerable research effort has been put into building sentence embeddings.

One of the first vector representations (embeddings), bag-of-words (BOW), is an intriguing approach in which the text is represented as a bag (multiset) of its words, with each word represented by its occurrence in the text (Parsing, 2009). The disadvantage of this strategy was that the embeddings were handcrafted, unlike the Word2Vec approach, which used a machine learning process to predict word embeddings (Mikolov et al., 2013). In Word2Vec, each word embedding is selected based on its overall context in the training corpus and can express the latent semantic of words. It automatically expresses the semantics of the whole sentences, though, so several approaches were proposed to tackle this problem. The most popular was

representing the sentence embedding as a weighted average of the sentence's word vectors. Because every word has the same embedding regardless of its meaning in the entire sentence, such an approach is not resistant to sentence changes and context semantics.

Bidirectional Encoder Representations from Transformers (BERT) is a very well known technique for constructing high-quality sentence embeddings that can express the dynamic and latent meaning of the whole sentences better than any previous approach. Its sentence embeddings can accurately reflect the meaning of the input text, making a significant difference in the quality of the downstream tasks performed. An even better variant of the BERT-based architecture, RoBERTa, has emerged and has lately become unquestionably state-of-the-art in terms of sentence embedding construction (Liu et al., 2019; Dadas et al., 2020).

2.2 Distance Metric Learning

Learning embeddings where instances from the same class are closer than examples from other classes is known as Distance Metric Learning (DML) (Qian et al., 2019). DML recently has drawn much attention due to its wide applications, especially in image processing. It can be used in the classification tasks together with the k-nearest neighbour algorithm (Weinberger and Saul, 2009), clustering along with K-means algorithm (Xing et al., 2002) and semi-supervised learning (Wu et al., 2020). DML's objective is to create embeddings similar to examples from the same class but different from observations from other classes. (Movshovitz-Attias et al., 2017). In contrary to the cross-entropy loss, which only takes care of intra-class distances to make them linearly separable, the DML approach maximizes inter-class and minimizes the intra-class distances (Wen et al., 2016). Aside from that, a typical classifier based solely on cross-entropy loss concentrates on class-specific characteristics rather than generic ones, as it is only concerned with distinguishing between classes rather than learning their representations. DML focuses on learning class representations, making the model more generalizable to new observations and more robust to outliers.

2.2.1 Contrastive Loss

Contrastive Loss is one of the methods in DML (Hadsell et al., 2006). It concentrates on pairs of similar and dissimilar observations, whose distances are attempted to be minimized if they belong to the same class and maximized if they belong to different classes. The loss function is given in Equation 1.

$$L\left(W,\left(Y,\overrightarrow{X}_1,\overrightarrow{X_2}\right)^i\right)=(1-Y)L_S\left(D_W^i\right)+YL_D\left(D_W^i\right) \tag{1}$$

where $\left(Y,\overrightarrow{X}_1,\overrightarrow{X}_2\right)^i$ denotes the labeled sample pair of with the index i, L_S represents the loss function for a pair of similar points, L_D is the loss function applied for pair of dissimilar points and D_W denotes distance function between pair of points $\overrightarrow{X_1},\overrightarrow{X_2}$.

2.2.2 Triplet Loss

Triplet Loss is similar to Contrastive Loss but works with triplets instead of pairs, is another solution to the DML problem (Schroff et al., 2015). Each triplet comprises an anchor, a positive, and a negative observation. Positive examples are members of the same class as an anchor, but negative instances belong to a separate class. Because it considers more observation simultaneously, it optimizes the embedding space better than Contrastive Loss. The actual formula for Triplet Loss is in Equation 2.

$$L=\sum_{i=1}^{N}\left[\|f_i^a-f_i^p\|_2^2-\|f_i^a-f_i^n\|_2^2+\alpha\right]_+ \tag{2}$$

where $f(x)$ represents the embedding that embeds ab observation x into a d-dimensional Euclidean space. x_i^a denotes an $anchor$, x_i^p $(positive)$ is the observation from the same class as the anchor, x_i^n $(negative)$ denotes an observation belonging to a different than the anchor class, α is an imposed between positive and negative pairs margin.

The most typical issue with triplets and contrastive learning is that as the number of observations in a batch grows, the number of pairs and triplets grows squarely or cubically. Another issue that might arise is the use of training pairs and triplets that are relatively easy to distinguish, leading to poor model generalization. Semi-solutions of the above problems are as introducing τ a temperature parameter that controls the separation of classes (Chen et al., 2020), or hard triples, which creates triplets based on harder negatives (Hermans et al., 2017).

2.2.3 ProxyNCA Loss

It is a more general approach to solving a problem with high resource consumption (Movshovitz-Attias et al., 2017). It employs proxies, artificial data points that represent the entire dataset. One proxy approximates one class; therefore, there are as many proxies as classes. This technique drastically reduces the number of triplets while simultaneously raising the convergence rate since each proxy make the triplet more resistant to outliers. The proxies are integrated into the model as trainable parameters since synthetic data points are represented as embeddings. Equation 3 depicts a ProxyNCA loss formula.

$$L = -\log\left(\frac{\exp\left(-d\left(\frac{x_i}{\|x_i\|_2}, \frac{f(x_i)}{\|f(x_i)\|_2}\right)\right)}{\sum_{f(z)\in Z}\exp\left(-d\left(\frac{x_i}{\|x_i\|_2}, \frac{f(z)}{\|f(z)\|_2}\right)\right)}\right) \tag{3}$$

where C_i is a set of observations from the same class, $f(a)$ denotes a proxy function returning class proxy for given parameter a, $|a||_2$ is the L^2-Norm of the vector a, $d\left(x_i, f\left(x_i\right)\right)$ denotes a distance between the sample x_i and proxy $f\left(x_i\right)$, Z denotes set of all proxies, where $f(z) \in Z$ and $z \notin C_i$.

A single proxy per class may not be enough to represent the class's inherent structure in real-world data. Another DML loss function has been created that introduces multiple proxies per class - SoftTriple Loss (Qian et al., 2019). ProxyNCA Loss can produce better embeddings while maintaining a smaller number of triplets than Triplet Loss or Contrastive Loss. The SoftTriple Loss is defined by the formulas in Equations 4 and 5.

$$\ell_{SoftTriple} = -\log\frac{\exp\left(\lambda\left(\mathcal{S}'_{i,y_i} - \delta\right)\right)}{\exp\left(\lambda\left(\mathcal{S}'_{i,y_i} - \delta\right)\right) + \sum_{j\neq y_i}\exp\left(\lambda\mathcal{S}'_{i,j}\right)} \tag{4}$$

$$\mathcal{S}'_{i,c} = \sum_k \frac{\exp\left(\frac{1}{\gamma}\mathbf{E}(\mathbf{x_i})^\top \mathbf{w}_c^k\right)}{\sum_k \exp\left(\frac{1}{\gamma}\mathbf{E}(\mathbf{x_i})^\top \mathbf{w}_c^k\right)}\mathbf{E}(\mathbf{x_i})^\top \mathbf{w}_c^k \tag{5}$$

where, C denotes the class number, k is the number of proxies representing observations from SoftTriple for each class, δ defines a margin between the example and class centers from different classes, λ reduces the influence from outliers and makes the loss more robust, γ is the scaling factor for the entropy regularizer, x_i defines the single observation represented as an array of tokens, $E(\cdot) \in \mathbf{R}^d$ indicates the encoder, $\mathbf{w}_c^k$ are weights representing proxy embeddings of the class c (there are k of them).

3 Our Approach

For fine-tuning pre-trained language models, we offer a novel objective function. It is based on the supervised cross-entropy loss and the SoftTriple Loss (Qian et al., 2019). The latter component is a loss from the Distance Metric Learning (DML) family of losses, which learns an embedding by capturing similarities between embeddings from the same class and distinguishing them from embeddings from different classes (Qian et al., 2019).

For a classification problem let us denote:

- N the number of observations,
- C the class number,
- y_{ic} the objective probability of the class c for the ith observation,
- β the scaling factor that tunes influence of both parts of the loss.

The novel goal function is given by the following formula:

$$\mathcal{L} = (\beta)\ell_{MCE} + (1-\beta)\ell_{SoftTriple} \tag{6}$$

, where

$$\ell_{MCE} = -\frac{1}{N}\sum_i^N\sum_c^C y_{ic}\log\left(p_{ic}\right) \tag{7}$$

It can be applied for different encoders $E(\cdot) \in \mathbf{R}^d$ from both image and natural language processing domains.

3.1 Model

In our work, we use the objective function from Equation 6 to fine-tune the pre-trained BERT-based language models provided by the *huggingface* library as RoBERTa-base and RoBERTa-large. In the standard setting, the single input text is first tokenized with WordPiece embeddings (Wu et al., 2016), which produces a vector of tokens x_i with a maximum length of 512, with $[CLS]$ at the beginning of an array, $[EOS]$ at the end and $[SEP]$ between tokens representing different sentences. The output of RoBERTa model $E(x_i) \in \mathbf{R}^d$ is an array of embeddings, where each input token has its corresponding embedding.

3.1.1 Multinominal Cross-Entropy Loss

In our experiments, we used the multinominal cross-entropy loss calculated in the same way as it was proposed by the authors of the BERT language model(Devlin et al., 2018). The sentence representation is obtained by pooling the output of the model $E(x_i) \in \mathbf{R}^d$ and passing it to the C dimensional single fully connected layer. Its output is passed to the softmax function generating probabilities p_{ic}, which are, along with objective probabilities y_{ic}, directly feeding the multinominal cross-entropy loss.

3.1.2 SoftTriple Loss

The second component of the TripleEntropy 6 is SoftTriple Loss (3), responsible for a more robust and better generalization of the model during tuning. It is fed by the direct output of the model $E(x_i) \in \mathbf{R}^d$, even before pooling. It means that if the batch size is B, then the total number of embeddings that feed SoftTriple Loss during one training iteration is $B * |x_i|$. This implementation ensures that the proxies representing each class will be well approximated so that the quality of fine-tuning increases.

Our implementation is a development of the earlier work (Gunel et al., 2020), where Contrastive Loss was applied only to the embedding corresponding to the first $[CLS]$ token of the input vector x_i. We apply SoftTriple Loss to the embeddings corresponding to all tokens from the input vector x_i, which ensures the better generalization of the fine-tuning process but requires more computing power. Fortunately, the SoftTriple Loss is significantly more efficient than the Contrastive Loss since it generates triplets not from all observations but its approximated proxies.

3.2 Training Procedure

Each result (average accuracy) was obtained as based on 4 seed runs (2, 16, 128, 2048), where each run was 5-fold cross-validated. It means that each accuracy result is an averaged of 20 different results. Apart from that, each result was based on the best parameter combination obtained by grid search which included parameters $k \in \{10, 100, 1000, 2000\}$, $\gamma \in \{0.01, 0.03, 0.05, 0.07, 0.1\}$, $\lambda \in \{1, 3, 3.3, 4, 6, 8, 10\}$, $\delta \in \{0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 1\}$ and $\beta \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. We noticed that for most experiments, the best hyperparameter set is following $k = 2000$, $\gamma = 0.1$, $\lambda = 3.3$, $\delta = 0.3$ and $\beta = 0.9$.

3.3 Datasets

We employed a variety of well-known datasets from SentEval (Conneau and Kiela, 2018) along with the IMDb (Maas et al., 2011) for model evaluations that covered both text classification and textual entailment as two important natural language tasks in order to assess the general use of TripleEntropy. Additionally, we have examined the performance of our method when the number of training examples is limited to 1,000 and 10,000 observations on sampled datasets. Table 1 shows the description of the datasets and their sampled versions.

4 Results

Results are presented in the form of comparison between the performance of the RoBERTa-base (RB) and the RoBERTa-large (RL) models as a baselines and the RoBERTa-base with SoftTriple Loss (RB SoftTriple) as well as RoBERTa-large with SoftTriple Loss (RL SoftTriple). Moreover, we have created 4 groups depending on the size of the dataset. In the first group, we present results regarding the small-sized datasets with the number of sentences of 1,000. In the second group, we explore results for the medium-sized datasets in which the number of sentences is not greater than 5,000 and not smaller than 4,000. In the third group, we present results belonging to the large-sized datasets with the number of sentences larger than 10,000 and fewer than 11,000. The extra-large-sized group consists of elements where the number of observations is larger than 50,000.

The RB baseline models were trained with the use of AdamW optimizer (Kingma and Ba, 2014), beginning learning rate 1e-5, L2 regularization, learning rate scheduler and linear warmup from 0 to 1e-5 for the first 6% of steps and batch size of 64. The RB SoftTriple models were trained on the same set of hyperparameters as the baseline models they refer to and additional parameters specific to SoftTriple Loss as it is described in Section 3.2.

4.1 RB SoftTriple for small datasets

Table 2 presents the results for the datasets containing 1,000 sentences. We observe that models trained using TripleEntropy have a higher performance than the baselines by about 0.78% on av-

Dataset	# Sentences	# Classes	Sampled subsets	Task
SST2	67k	2	10k, 1k	Sentiment (movie reviews)(Socher et al., 2013)
IMDb	50k	2	10k, 1k	Sentiment (movie reviews) (Maas et al., 2011)
MR	11k	2	1k	Sentiment (movie reviews) (Pang and Lee, 2005)
MPQA	11k	2	1k	Opinion polarity (Wiebe et al., 2005)
SUBJ	10k	2	1k	Subjectivity status (Pang and Lee, 2004)
TREC	5k	6	1k	Question-type classification (Pang and Lee, 2005)
CR	4k	2	1k	Sentiment (product review) (Hu and Liu, 2004)
MRPC	4k	2	1k	Paraphrase detection (Dolan et al., 2004)

Table 1: SentEval and IMDb datasets, and their sampled subsets, used for our evaluation.

erage. It is worth noting that the gain in performance is observed at each dataset, especially for the TREC-1k and MRPC-1k, where it amounts to 2.29% and 1.11%, respectively.

4.2 RB SoftTriple for medium datasets

Table 3 shows the results based on the datasets containing more than 1,000 sentences and less than 11,000. Here, we can observe that models trained using TripleEntropy have higher performance than the baselines by about 0,86% on average. The highest gain in performance is observed in the case of TREC and MRPC datasets by 1.00% and 1.28%, respectively.

4.3 RB SoftTriple for large datasets

Table 4 shows the results based on the datasets containing 10,000-11,000 sentences. The gain in the performance amounts 0.20%.

4.4 RB SoftTriple for extra-large datasets

Table 5 shows the results based on the datasets containing more than 50,000 sentences. The gain in the performance is not as high as in the case of the medium and small-sized datasets, and it is 0.04% on average, which is not significant.

4.5 RL SoftTriple for small datasets

We have compared our results to the related work (Gunel et al., 2020) where the authors claim the performance gains over baseline RoBERTa-large by applying loss function consisted of cross-entropy loss and Supervised Contrastive Learning loss. The work shows the improvement over baseline in the few-shot learning defined as fine-tuning based on the training dataset consisted of 20, 100 and 1,000 observations. In order to compare our new loss function with the results from the related work we conducted experiments where the baseline was RoBERTa-large (RL) with cross-entropy loss and compared it to the RoBERTa-large with cross-entropy and SoftTriple loss (RL SoftTriple) on the dataset consisted of 1,000 observations. We can observe that our method yields a gain over baseline of 0.48%, which is higher than the performance improvement over baseline for a dataset of the same size from related work, whose improvement over baseline is 0.27%. The results are presented on the table 6.

4.6 Discussion

We can observe that our method improves the performance most significantly for the small-sized dataset by 0.87% in the case of RoBERTa-base baseline and 0.48% in the case of RoBERTa-large baseline and the medium-sized dataset, where the increase amounts to 0.86%. For the large-sized dataset, the increase over baseline is 0.20%, while for the extra-large-sized dataset, the gain over baseline amounts 0.04%. Our experiments show consistent performance improvement over baseline when using SoftTriple loss, which is highest for the small and medium-sized datasets and decreases for the large and extra-large sized datasets. It is an improvement over previous related work, where the performance improvement for the supervised classification tasks was achieved only for the few-shot learning settings (Gunel et al., 2020).

We also conclude that the smaller the dataset is, the higher our new goal function's performance gain over baseline. This observation is consistent with the conclusions of previous work (Gunel et al., 2020). When the dataset is larger than about 10k observations, the gain is negligible. In addition, our work focuses on datasets of no less than 1k observations, so we do not know how it behaves in case of few-shot learning, which in contrast has

Model	SST2-1k	IMDb-1k	SUBJ-1k	MPQA-1k	MRPC-1k	TREC-1k	CR-1k	MR-1k
RB	88.63	81.00	94.61	87.75	78.01	79.80	91.57	85.89
RB ST	89.09	81.45	94.70	87.93	79.12	82.09	92.16	86.39

Table 2: RoBERTa-base (RB) vs RoBERTa-base with SoftTriple Loss (RB ST) for small sized datasets

Model	MRPC	TREC	CR	MR
RB	83.11	96.19	93.28	89.09
RB SoftTriple	84.39	97.19	93.58	89.29

Table 3: RoBERTa-base (RB) vs RoBERTa-base with SoftTriple Loss for medium sized datasets

Model	SST2-10k	IMDb-10k	SUBJ	MPQA
RB	92.63	85.12	96.83	91.08
RB SoftTriple	92.79	85.23	97.15	91.30

Table 4: RoBERTa-base (RB) vs RoBERTa-base with SoftTriple Loss for large sized datasets

Model	SST2	IMDb
RB	94.89	87.10
RB SoftTriple	94.95	87.12

Table 5: RoBERTa-base (RB) vs RoBERTa-base with SoftTriple Loss for extra large sized datasets

Model	SST2-1k	MPQA-1k	MRPC-1k	TREC-1k	CR-1k	MR-1k
RL	91.96	90.18	76.09	83.75	93.43	89.69
RL SoftTriple	92.14	90.59	77.16	84.59	93.62	89.89

Table 6: RoBERTa-large (RB) vs RoBERTa-large with SoftTriple Loss for small sized datasets

been well documented in the case of work (Gunel et al., 2020). The performance comparison between baseline and our method throughout dataset size is depicted in Figure 1.

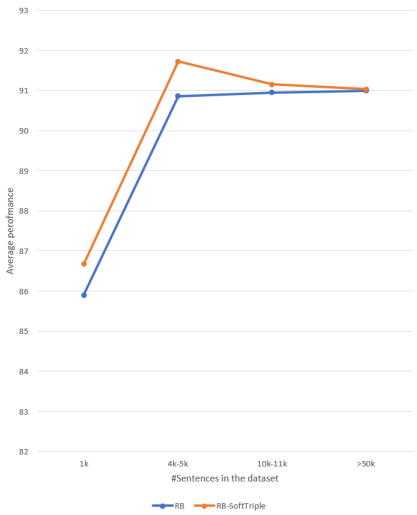

Figure 1: Performance comparison between RB and RB SoftTriple

5 Conclusions

We proposed a supervised Distance Learning Metric objective that increases the performance of the RoBERTa-base models, which are strong baselines in the Natural Language Processing tasks. The performance is proved over multiple tasks from the single sentence classification and pair sentence classification to be higher by about (0.02%-2.29%) depending on the training dataset size. In addition, each result has been confirmed through tests with 5-fold cross-validation on 4 different seeds to increase its reliability. In the future, we plan to extend the application of our method to compare the results with language models from different architectures to investigate its general usefulness in other tasks.

References

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *In-*

ternational conference on machine learning, pages 1597–1607. PMLR.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

Sławomir Dadas, Michał Perełkiewicz, and Rafał Poświata. 2020. Pre-training polish transformer-based language models at scale. In *International Conference on Artificial Intelligence and Soft Computing*, pages 301–314. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

William Dolan, Chris Quirk, Chris Brockett, and Bill Dolan. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. *arXiv preprint arXiv:2011.01403*.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Alexander Hermans, Lucas Beyer, and Bastian Leibe. 2017. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

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

Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. 2017. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 360–368.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint cs/0409058*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*.

Constituency Parsing. 2009. Speech and language processing.

Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. 2019. Softtriple loss: Deep metric learning without triplet sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6450–6458.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Kilian Q Weinberger and Lawrence K Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2).

Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. 2016. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer.

Lyndon White, Roberto Togneri, Wei Liu, and Mohammed Bennamoun. 2015. How well sentence embeddings capture meaning. In *Proceedings of the 20th Australasian document computing symposium*, pages 1–8.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2):165–210.

Sanyou Wu, Xingdong Feng, and Fan Zhou. 2020. Metric learning by similarity network for deep semi-supervised learning. In *Developments of Artificial Intelligence Technologies in Computation and Robotics:*

Proceedings of the 14th International FLINS Conference (FLINS 2020), pages 995–1002. World Scientific.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Eric Xing, Michael Jordan, Stuart J Russell, and Andrew Ng. 2002. Distance metric learning with application to clustering with side-information. *Advances in neural information processing systems*, 15:521–528.