EasyChair Preprint

№ 15448

An Approach for Evaluating Semantic Similarity in Research Papers via Siamese BERT Architecture

Pritam Sarkar, Soumyaneel Sarkar, Wazib Ansar and
Amlan Chakrabarti

November 20, 2024

An Approach for Evaluating Semantic Similarity in Research Papers via Siamese BERT Architecture

Pritam Sarkar[1], Soumyaneel Sarkar[1], Wazib Ansar[1], and Amlan Chakrabarti[1]

A.K. Choudhury School of Information Technology, University of Calcutta, Kolkata, India
sarkarpritam0007@gmail.com, soumyasarkar309@gmail.com, wazibansar@ymail.com, acakcs@caluniv.ac.in

Abstract. Document similarity analysis is critical for various NLP tasks like information retrieval and plagiarism detection. Traditional methods based on word-to-word mapping struggle with capturing contextual nuances. Existing solutions lack the capability to provide domain-specific accuracy and enriched search experiences. One such field is finding similar research papers. Often researchers struggle to find papers similar to a certain paper and have to rely on basic keyword-based search. This hinders to provide the best match based on the overall context. In this work, we propose a novel methodology that integrates BERT with a Siamese Neural Network to capture semantic textual similarity of research papers. Our approach goes beyond simple similarity evaluation by conducting a nuanced semantic search of overall context and provides a representative similarity score. This offers a more accurate and refined search experience. Furthermore, we curate a dataset of over 10,000 NLP research paper abstracts to train our model. The model excels in identifying the contextual relationships between documents, making it highly effective for domain-specific applications. This model can significantly improve the user experience in document retrieval systems, particularly for academic research and recommendation.

Keywords: BERT · Data Science · NLP · Semantic Similarity · Siamese Neural Network.

1 Introduction

Semantic textual similarity plays a critical role in a wide range of Natural Language Processing (NLP) applications, including information retrieval, plagiarism detection, and document classification [1]. Traditional methods, which typically rely on string matching and fingerprinting techniques, excel in identifying verbatim and copy-paste text similarity but struggle with the more complex task of detecting paraphrased and semantically similar content [2]. As paraphrased sentences retain the same meaning while varying in structure and word choice, extracting meaningful semantic information from them poses significant challenges.

Recent advancements in NLP have shifted towards leveraging deep learning models and Large-Language Models (LLMs) [3] to capture semantic meaning more effectively. Notably, transformer-based[4] models like Bidirectional Encoder Representations from Transformers (BERT)[5], [6] have shown great promise in contextual understanding by utilizing bi-directional embeddings. This approach enables models to generate context-aware embeddings for both original and paraphrased texts, thus improving the detection of semantic similarity [7], [8].

Previous works have employed Siamese networks [9] [10] to tackle similarity metrics on variable-length sequences, as demonstrated by Neculoiu et al. [11] in the context of job title matching. Lo [12] further explored fine-tuning BERT for semantic textual similarity across two languages. Additionally, Viji and Revathy [13] introduced a hybrid model combining a fine-tuned BERT with a Siamese Bi-LSTM, improving the accuracy and robustness of similarity predictions.

However, data scarcity and the challenge of fully capturing the complex semantic relationships present in academic texts remain [14]. While conventional techniques such as cosine similarity are prevalent, they often fall short in nuanced academic contexts. To address these challenges, we propose a novel architecture that combines BERT with Siamese networks, fine-tuned on academic datasets. This approach aims to transcend the limitations of traditional methods by capturing deeper semantic relationships, offering improved accuracy in identifying contextual similarity between academic documents. Our principal contributions are as follows:

1. We propose a novel architecture that combines BERT with Siamese networks to find similarity among research papers.
2. To train the model, we curate a dataset of over 10,000 research paper abstracts.
3. The proposed model excels in determining contextual similarity between academic documents and quantifying them with a similarity score.

This paper has been organized as in the following manner. Section 2 explores the related works in the domain. Section 3 describes the proposed methodology. Section 4 states the experimental setup. Section 5 presents the results while Section 6 discusses the results and its applicability. Finally in Section 6, this paper is concluded.

2 Related Works

Conventional methods often struggle with capturing contextual and word-order information, leading to data sparsity and scalability issues.To address these limitations, researchers have explored deep learning techniques. Neculoiu et al [11]. proposed a Siamese Recurrent Neural Network (RNN)[15] [16] architecture to learn similarity metrics on variable-length character sequences, such as job titles, demonstrating effectiveness with limited supervision. Building on these advancements, Lo [12] discusses fine-tuning BERT, a popular transformer-based model, for semantic textual similarity across two languages. The use of the Semantic

Textual Similarity (STS) benchmark dataset highlights this approach's effectiveness. While Viji and Revathy (2022) [13] introduce a hybrid approach combining Weighted Fine-Tuned BERT extraction with a deep Siamese Bi-LSTM model, leveraging the strengths of both to enhance text similarity [17] predictions' accuracy and robustness. Additionally, successful applications of Siamese networks[18] in document verification, such as writer-independent signature verification and authorship verification, motivate further exploration for academic document similarity evaluation (Dey et al., 2017; Sun et al., 2021) [19]. Ansar et al. [7] put forth a unique pictorial representation technique for text utilizing BERT embeddings which were potent enough to be utilized for comparing texts based on image comparison metrics. Later, they devised transformer encoder architecture for generating pictorial representations together with a siamese transformer architecture to efficiently determine similarity of unequal texts [8]. Data scarcity remains a challenge in document similarity tasks. While traditional methods like cosine similarity [20] are widely used, they may not fully capture the intricate semantic relationships in academic language. Our work aims to transcend these limitations by leveraging transformer models fine-tuned on comprehensive academic datasets.

3 Proposed Methodology

3.1 Overview

The system leverages advanced natural language processing (NLP) techniques to efficiently compare and analyze academic document abstracts. It begins with the collection and curation of a substantial dataset of academic documents, sourced from the ACL Anthology Corpus. The data is cleaned, handling missing values, removing duplicates, and correcting errors. Key fields are extracted, and new fields for paraphrased content and similarity scores are added. The curated dataset is saved and managed securely on platforms like Kaggle.

The text data is then transformed into numerical vector embeddings using techniques like BERT. A Siamese neural network [18] is then trained for semantic textual similarity tasks, comparing abstracts within the dataset. Incoming abstracts undergo the same preprocessing and embedding steps for consistency. The system computes similarity scores between new and existing abstracts, providing concise, relevant matches. This aids researchers in identifying highly relevant literature, facilitating efficient literature review and research advancement in computational linguistics and related fields. The detailed methodology has been presented herein-below.

3.2 Dataset Curation

The process begins with the dataset stage, where a substantial collection of academic documents, including research papers, journal articles, and other scholarly works from various domains, is assembled. In this paper we put forth a new

dataset to determine similarity among abstracts of research papers on NLP. It contains 10,072 abstracts of papers , parapharased version of abstracts along with the corresponding similarity score. This dataset provides the raw material necessary for subsequent analysis and modeling. The steps in curation of dataset are as follows:

Step 1: Data Collection
- i Collected the ACL Anthology Corpus from HuggingFace[1].
- ii Processed the dataset to obtain a CSV file with abstracts.

Step 2: Data Pre-processing
- i Handled missing values based on the 'abstract' field.
- ii Removed duplicates.
- iii Corrected errors and inconsistencies.

Step 3: Abstract Similarity Detection Dataset Creation
- i Extracted the following fields: ['acl_id', 'abstract', 'corpus_paper_id', 'url', 'numcitedby', 'title', 'url', 'year'].
- ii Created a new dataset with only these columns.
- iii Renamed the 'abstract' column to 'original'.
- iv Added a new field named 'paraphrased' containing paraphrased abstract. The contents have been generated using 'gemini-pro'[21].
- v Annotated the data for supervised learning, where 'original' and 'paraphrased' are the input variables and 'similarity_score' is the target variable. The 'similarity_score' field is populated using 'gemini-pro' by comparing 'original' and 'paraphrased'.

This algorithm processes a dataset by focusing on a specific set of fields and modifying its structure. After selecting and creating a new dataset, the `abstract` column is renamed to `original` to better reflect the content. Using `gemini-pro`, the algorithm generates a paraphrased version of the `original` content and calculates a similarity score to compare the two. Both the paraphrased text and the similarity score are stored in newly created columns. This process enhances the dataset with enriched information for further analysis.

3.3 Text Embeddings

The text data is converted into vector embeddings using advanced NLP techniques like BERT[5]. This process transforms the text into numerical representations that capture semantic and contextual information. After preprocessing, the data is passed to a pre-trained transformer [22] model and tokenizer embedding function. This utilizes models such as BERT to create embeddings that reflect the contextual meaning of the documents, translating them into a high-dimensional vector space where similar documents are positioned closer together. The tokenizer converts the text into tokens for the transformer model, enabling effective text processing.

[1] https://huggingface.co/datasets/WINGNUS/ACL-OCL

3.4 Model Creation

The vector embeddings are then utilized to train a Deep Learning (DL) model, which is a Siamese neural network architecture for semantic textual similarity tasks. The DL model is trained using techniques like backpropagation and gradient descent to learn the underlying patterns and relationships between abstract pairs. The model has been described step by step as follows:

1. **Input Abstracts:** Two abstracts are provided as input: the Original Abstract and the Similar/Dissimilar Abstract.
   (a) **Tokenization:**
      i. Feed both the original and paraphrased abstracts into a tokenizer.
      ii. The tokenizer converts the textual data from both abstracts into numerical tokens that a machine learning model can process.
      iii. Store the resulting tokenized versions of both the original and paraphrased abstracts for further analysis.

   The tokenization process transforms raw textual data into numerical tokens that machine learning models can process. First, both the original and paraphrased abstracts are fed into a tokenizer, which breaks down the text into smaller components, assigning numerical values to each token. This conversion is essential for enabling the model to interpret the text and perform tasks such as similarity analysis or classification on the tokenized data.
2. **Embedding with Model Transformer:**
   (a) The tokenized abstracts are passed through a Model Transformer (e.g., BERT).
   (b) The Model Transformer processes the tokens and generates dense vector representations (embeddings) for each abstract.
   (c) The output is Original Abstract embeddings and Similar Abstract embeddings.
3. **Pooling:**
   (a) The embeddings are then passed through a Pooling layer.
   (b) Pooling summarizes the embeddings into a fixed-size vector, usually by taking the mean or maximum value of the embeddings.
   (c) The result is a condensed representation of the abstracts.
4. **Sentence Transformer:**
   (a) The pooled embeddings are fed into a Sentence Transformer.
   (b) The Sentence Transformer further refines the embeddings to capture the sentence-level semantics.
5. **Linear, Layer Norm, Leaky ReLU Activation and Dropout:**
   (a) The Linear layer applies a linear transformation to the incoming data:

$$y = xA^T + b \tag{1}$$

   where A is the weight matrix and b is the bias.

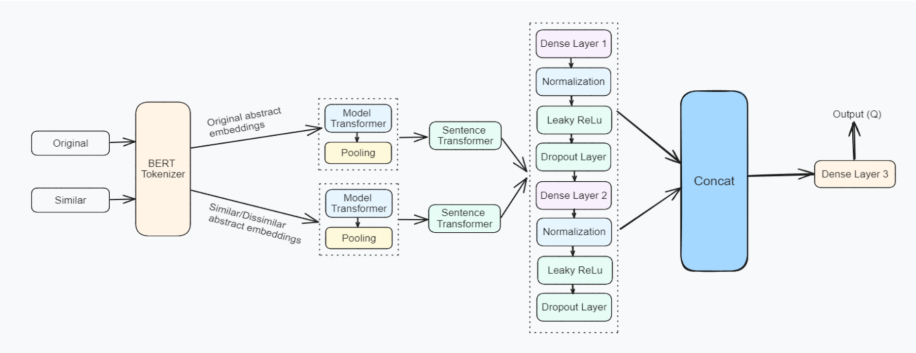

Fig. 1. Model architecture.

(b) LayerNorm normalization technique normalizes the inputs across the features for each data point in the mini-batch:

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{2}$$

where μ is the mean, σ^2 is the variance, and ϵ is a small constant added for numerical stability.

(c) A Leaky ReLU activation function is applied to introduce non-linearity:

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{if } x < 0 \end{cases} \tag{3}$$

where α is a small positive constant that allows for a small gradient when x is negative.

(d) A Dropout layer is used to prevent overfitting by randomly setting a fraction of the input units to zero during training:

$$\text{Dropout}(x) = x \cdot \text{Bernoulli}(p) \tag{4}$$

where p is the probability of keeping a unit active.

6. **Concat:**
   (a) Concatenation is used to concatenate the incoming original and similar tensors along a specified dimension. This is useful for combining tensors along a particular axis to form a larger tensor.
7. **Similarity Score:**
   (a) The final Similarity Score quantifies the degree of similarity between the Original Abstract and the Similar Abstract.
   (b) This score can be used to assess the originality of the given input abstract relative to the trained ACL dataset model.

4 Experimental Setup

The experimental setup is critical to the validity and reproducibility of our research, outlining the methodologies, procedures, and tools used to investigate our research questions. This setup ensures that our investigation is conducted with rigor and transparency. Our setup includes a detailed step-by-step process for evaluating the functionality of our integrated model and tool. Initially, users authenticate and log into the system, where they are directed to the chat page. Here, they can input written text or upload an image of an abstract, which is then transcribed into text by our system. Upon submission, the abstract is sent to our backend server through an API call. The backend architecture comprises two servers. The first server, connected to the Next.js frontend, handles authentication and forwards the abstract query to a Flask-based server. This Flask server performs similarity checks using a MongoDB database (current version) for article details and embeddings, and calculates cosine similarity scores. The model is trained on a system equipped with a GPU P100 to enhance computational efficiency and uses the latest version of PyTorch for model operations. Additionally, development and testing were conducted on an Asus laptop, ensuring portability and accessibility. This robust experimental setup allows users to seamlessly evaluate the integrity and originality of abstracts, leveraging advanced computational resources and thorough backend processing. Through this meticulous setup, we ensure the reproducibility and validity of our study, inviting further exploration and advancement in our field.

Dependencies

- Flask
- numpy
- pandas

- google-generativeai
- pymongo
- sentence_transformers
- torch
- tqdm
- transformers
- bert_score
- datasets
- scipy
- Scikit-learn

Database

MongoDB

GPU

GPU P100

System

- Asus Vivobook 16: 16GB, Ryzen 7 5800HS
- Asus F Dash 15: 16GB, Intel i5 12500H

Application Developed in

Next.js, Node.js, TypeScript, Tailwind CSS.

Hyperparameters

The details of the hyperparameters used are as follows:

Table 1. Model Parameters

Parameter	**Value**
Epochs	8
Learning Rate	2e-6
Optimizer	Adam
Max Length	212
Weight Decay	1e-5
Loss Function	MSELoss

5 Results

By leveraging the power of BERT for embeddings and a Siamese network for comparing abstracts, the proposed methodology determines the similarity between abstracts effectively. This can be observed from the results obtained in Table 2 where the efficacy has been demonstrated in terms of various metrics. This shows that the transformer encoder-decoder[23] model, with its self-attention and attention mechanisms, captures long-range dependencies between words, making it a powerful tool for machine translation[24] and other NLP tasks, ensuring accurate translation and other applications.

Mean Squared Error (MSE): The average of the squared differences between predicted and actual values. It emphasizes larger errors.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Table 2. Test Results Summary

Metric	Value
Avg. Test Loss	0.0010
Mean Absolute Error	0.0209
Mean Squared Error	0.0010
R-squared	0.7067
Pearson Correlation	0.8706
Spearman's Rank Correlation	0.6564
Testing Time	23s

Mean Absolute Error (MAE): The average of the absolute differences between predicted and actual values. It treats all errors equally.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of data points.

In our approach, y_i is the similarity score between two sentences in our dataset and $\hat{y}_i$ is the predicted similarity score that the model is predicting by comparing two sentences.

Calculation of MSE & MAE: MSE is calculated by taking the average of the squared difference between the model's predicted similarity scores ($\hat{y}_i$) and the similarity scores from the dataset (y_i).

MAE is calculated by taking the average of the absolute difference between the model's predicted similarity scores ($\hat{y}_i$) and the similarity scores from the dataset (y_i).

Table 3 compares various methods for measuring similarity between text pairs using three key metrics: Cosine Similarity, BERTScore, and Our Approach. Across all three examples, our approach demonstrates competitive or superior performance. Specifically, in the second and third rows, where the comparison involves identical or closely related text pairs, our approach scores the highest possible value (100) and 90.12, respectively, showcasing its robustness in capturing fine-grained similarities. The second row shows a perfect agreement across all metrics, indicating that all approaches, including ours, can effectively handle simple, repetitive statements. However, the third row presents a more complex semantic task, where "Our Approach" outperforms BERTScore (90.12 vs. 84.63), indicating its strength in understanding nuanced text relationships, even when other sophisticated models slightly under perform. In the first row "Our Approach" outperformed other metrics that is showcased in the table where Cosine Similarity of two dissimilar text pair is very high and our approach handles the test case very well.

Table 3. Comparison of different similarity scores and approaches.

Source	Incoming	Cosine Similarity	BERTscore	Our Approach
Despite the growing cultural presence of eSports, no corpus contains this genre of entertainment..	Football is a family of team sports that involve, to varying degrees, kicking a ball to score a...	81.71	39.03	31.05
Small, manually assembled corpora may be available for less dominant languages and dialects...	Small, manually assembled corpora may be available for less dominant languages and dialects...	100	99.5	100
During recent years there has been an increased interest to acquire or extend, on a large-scale, ...	Estimating the semantic similarity between text data is one of the challenging and open..	95.48	84.63	90.12

6 Discussion

The computed similarity scores provide insights into the contextual similarity between the incoming document abstract and the ACL abstracts. This analysis helps understand the degree of resemblance or relevance between the new abstract and existing literature in the ACL repository. In the evaluation stage, these similarity scores are analyzed to determine the degree of contextual similarity between the input document and existing documents, helping to identify documents that are highly similar or potentially overlapping with the input document.

Conclusion

This paper introduces an effective methodology to determine similarity of research paper abstracts using a BERT-based siamese neural network. Furthermore, a dataset has been specially curated from the ACL Anthology Corpus to perform this task. This streamlined methodology ensures efficient processing of abstracts, accurate similarity assessment, and meaningful results presentation, facilitating knowledge discovery and research advancement in computational linguistics and related fields. Supported by advanced transformer models and a robust evaluation mechanism, this comprehensive workflow empowers researchers to assess the contextual similarity of academic documents effectively,

ensuring the originality and uniqueness of their contributions within the academic landscape. In the future, we wish to extend dataset adding more samples to it. Another area for extension can be to customize the methodology make it suitable for deployment on edge devices.

Acknowledgments. We would like to thank Dr. Saptarsi Goswami, Assistant Professor, Bangabasi Morning College for his guidance and motivation. Apart from that, the author(s) received no financial support for the research, authorship, and/or publication of this article.

References

1. Chandrasekaran, Dhivya, and Vijay Mago. "Evolution of semantic similarity—a survey." ACM Computing Surveys (CSUR) 54, no. 2 (2021): 1-37.
2. Wang, Jiapeng, and Yihong Dong. "Measurement of text similarity: a survey." Information 11, no. 9 (2020): 421.
3. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. Advances in neural information processing systems 35, 22199–22213 (2022) Authorship Verification MDPI.
4. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems 30 (2017)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Koroteev, Mikhail V. "BERT: a review of applications in natural language processing and understanding." arXiv preprint arXiv:2103.11943 (2021).
7. Ansar, Wazib, Saptarsi Goswami, Amlan Chakrabarti, and Basabi Chakraborty. "TexIm: A Novel Text-to-Image Encoding Technique Using BERT." In Computer Vision and Machine Intelligence: Proceedings of CVMI 2022, pp. 123-139. Singapore: Springer Nature Singapore, 2023.
8. Ansar, Wazib, Saptarsi Goswami, and Amlan Chakrabarti. "TexIm FAST: Text-to-Image Representation for Semantic Similarity Evaluation using Transformers." arXiv preprint arXiv:2406.04438 (2024).
9. Ferreira, Anselmo, Nischay Purnekar, and Mauro Barni. "Ensembling shallow siamese neural network architectures for printed documents verification in data-scarcity scenarios." IEEE Access 9 (2021): 133924-133939.
10. Dey, Sounak, Anjan Dutta, J. Ignacio Toledo, Suman K. Ghosh, Josep Lladós, and Umapada Pal. "Signet: Convolutional siamese network for writer independent offline signature verification." arXiv preprint arXiv:1707.02131 (2017).
11. Neculoiu, Paul, Maarten Versteegh, and Mihai Rotaru. "Learning text similarity with siamese recurrent networks." In Proceedings of the 1st Workshop on Representation Learning for NLP, pp. 148-157. 2016.
12. Lo, Chi-kiu, and Michel Simard. "Fully unsupervised crosslingual semantic textual similarity metric based on BERT for identifying parallel data." In Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), pp. 206-215. 2019.
13. Viji, D., and S. Revathy. "A hybrid approach of Weighted Fine-Tuned BERT extraction with deep Siamese Bi–LSTM model for semantic text similarity identification." Multimedia tools and applications 81, no. 5 (2022): 6131-6157.

14. Kovács, György, Pedro Alonso, and Rajkumar Saini. "Challenges of hate speech detection in social media: Data scarcity, and leveraging external resources." SN Computer Science 2, no. 2 (2021): 95.
15. Zhou, Jie, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. "Deep recurrent models with fast-forward connections for neural machine translation." Transactions of the Association for Computational Linguistics 4 (2016): 371-383.
16. Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014).
17. Gomaa, Wael H., and Aly A. Fahmy. "A survey of text similarity approaches." international journal of Computer Applications 68, no. 13 (2013): 13-18.
18. Embarcadero-Ruiz, Daniel, Helena Gómez-Adorno, Alberto Embarcadero-Ruiz, and Gerardo Sierra. "Graph-based siamese network for authorship verification." Mathematics 10, no. 2 (2022): 277.
19. Dey, Sounak, Anjan Dutta, J. Ignacio Toledo, Suman K. Ghosh, Josep Lladós, and Umapada Pal. "Signet: Convolutional siamese network for writer independent offline signature verification." arXiv preprint arXiv:1707.02131 (2017).
20. Rahutomo, Faisal, Teruaki Kitasuka, and Masayoshi Aritsugi. "Semantic cosine similarity." In The 7th international student conference on advanced science and technology ICAST, vol. 4, no. 1, p. 1. South Korea: University of Seoul, 2012.
21. Team, Gemini, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut et al. "Gemini: a family of highly capable multimodal models." arXiv preprint arXiv:2312.11805 (2023).
22. Attali, Y., Runge, A., LaFlair, G.T., Yancey, K., Goodwin, S., Park, Y., Von Davier, A.A.: The interactive reading task: Transformer-based automatic item generation. Frontiers in Artificial Intelligence 5, 903077 (2022)
23. Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. CoRR, abs/1406.1078, 2014.
24. Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc V. Le. Massive exploration of neural machine translation architectures. CoRR, abs/1703.03906, 2017.