ALBERT Ensemble with Linguistic Knowledge Boosting for the SQuAD 2.0 Challenge

Stanford CS224N Default Project

Dane M. Hankamer
Department of Computer Science
Stanford University
dhank@stanford.edu

David J. Liedtka III
Department of Computer Science
Stanford University
dliedtka@stanford.edu

1 Research paper summary

Title	ALBERT: A Lite BERT for Self-supervised Learning of Language Representations
Venue	International Conference on Learning Representations (ICLR)
Year	2020 (conference paper)
URL	https://arxiv.org/abs/1909.11942

Table 1: Bibliographical information for ALBERT research paper (v6) [1].

Background. Transfer learning models such as ALBERT and BERT are two highly successful approaches toward general and robust natural language understanding systems. ALBERT, like BERT, constructs a multi-layer bidirectional transformer encoder that uses multiple unsupervised learning objectives for NLP tasks ranging from question answering to natural language inference. The ALBERT model attempts to improve the well-known BERT model by significantly reducing the number of parameters through decomposing embedding parameters into smaller matrices and sharing parameters across layers, thereby reducing the training time and learning fewer parameters. This work is monumentally important because it addresses the scalability problem that many state-of-the-art models experience when training hundreds of millions or billions of parameters. Moreover, ALBERT actually improves the state-of-the-art for our SQuAD 2.0 question answering task. ALBERT moves us one step closer to an NLP system that can answer questions about any text document from textbooks to financial reports.

Summary of contributions. This paper makes 3 main improvements to the previously state-of-the-art BERT model. First, ALBERT factorizes the embedding parameters by projecting the one-hot WordPiece embeddings (size E) and hidden layer embeddings (size H) into a lower dimensional embedding space of size E. This lower dimensional representation is then projected into hidden space H. Previously, BERT kept E and H the same, which intuitively seems suboptimal as WordPiece embeddings learn context-independent representations and hidden-layer embeddings learn context-dependent representations. Given the strength of BERT lies mostly in discovering the context-dependent representations, the factorization of ALBERT allows us to increase H ($H \gg E$ in some situations) and reduce the embedding parameters from $O(V \times H)$ to $O(V \times E + E \times H)$. Secondly, ALBERT allows parameter sharing across all layers. This greatly stabilizes the network parameters from layer to layer when compared to BERT. Thirdly, ALBERT implements inter-sentence modeling using a loss function based primarily on coherence instead of both topic prediction and coherence. Because the model already uses masked language modeling (MLM) loss, a second sentence-order prediction (SOP) loss function focuses solely on modeling inter-sentence coherence. In turn, ALBERT models perform better on multi-sentence encoding tasks than BERT implementations using just MLM loss (or MLM loss plus the recently eliminated next-sentence prediction (NSP) loss).

Limitations and discussion. While the best performing model in this paper, ALBERT-xxlarge, significantly improves performance over BERT-large across a myriad of downstream tasks with only 70% of the parameters, it iterates through the data about 3 times slower due to its massive structure. On the other hand, ALBERT-large outperforms similarly structured BERT-large while iterating through the data 1.7 times *faster*. This structure size limitation between ALBERT-xxlarge and ALBERT-large presents the challenge of applying methods such as block attention and sparse attention to ALBERT-xxlarge in order to increase the inference speed. Another meaningful discussion is the switch from NSP to SOP to increase document level understanding. Given the convincing evidence that SOP is an improvement over NSP, there might be more self-supervised training losses such as individual word or character ordering that could lead to additional representation power. The ALBERT paper also claims that removing Dropout significantly improves MLM accuracy. While there is limited empirical and theoretical evidence that Dropout can be harmful to transformer-based models, an interesting experiment would be to pinpoint the reasoning behind this phenomenon. Lastly, the ALBERT paper did not analyze individual errors across the numerous NLP tasks. It would be helpful to see what types of errors ALBERT tends to make when compared to the other general natural language understanding systems. Overall, we are convinced that ALBERT is the new state-of-the-art for many NLP tasks, to include our SQuAD 2.0 question answering task. The ALBERT model is proven as a top performer over many datasets (Wikipedia, BookCorpus, XLNet data, RoBERTa data, SQuAD, RACE, GLUE, etc.) and a wide range of NLP tasks, thereby making it arguably the best overall base NLP model today.

Why this paper? We chose this paper because it represents the current state-of-the-art for our SQuAD 2.0 question answering task. The question answering topic has always been an interest for both of us, especially given the commercial applications for better understanding complex legal documents, textbooks, or gleaning information from a lengthy paper by asking an NLP system a series of questions. After reading the paper, we believe we gained an intuition behind the design changes of ALBERT versus BERT, which is exactly what we were hoping for. That being said, we also read the BERT paper for a better understanding of transformer-based models [2].

Wider research context. ALBERT does a fantastic job of focusing on building a better representation of language as a whole. It is not focused on a singular task such as question answering, but instead is a technique for natural language processing pre-training that can be used for many downstream tasks. ALBERT, like BERT, uses bidirectional training to learn word context based on the surrounding words versus just the previous or next word. This recent breakthrough has massive implications for how we represent language based on context. Furthermore, additional tweaks such as SOP loss spark further research into language representation, structure, and document-level understanding of language by NLP systems. We already see overall language-level improvements in areas such as Google Search, which now uses a BERT-based system approximately 10% of the time to deal with nuances and relative importance of each word in the search query. State-of-the-art NLP systems are still far from perfect, but ALBERT makes a significant contribution to the broader story of NLP research.

2 Project description (1-2 pages)

Goal. Our goal is to improve the current state-of-the-art ALBERT model for the SQuAD 2.0 question answering task through ensembling and linguistic knowledge methods. Specifically, we hope to adopt the op-for-op re-implementation of the ALBERT system from the research paper above, develop boosted probabilities for certain text spans based on the question (i.e. looking for “’s" for questions beginning with “Whose"), and implement ensembling using multiple ALBERT models and the default BiDAF model. This work is important because ALBERT is a relatively new transformer-based system and is not perfectly tuned for many specific NLP tasks. That being said, many ensembling configurations, hyperparameter tuning, and linguistic knowledge boosters have not been tested in conjunction with ALBERT. Introducing multiple improvements to the current baseline ALBERT question answering system would be incredibly challenging and fulfilling for us, as well as potentially providing insights for question answering machines as a whole. Time permitting, we will also explore a data augmentation technique such as SwitchOut or a random synonym-based word swap (using NLTK) on the contexts of the SQuAD 2.0 dataset and train one of our ALBERT models on this augmented dataset [3].

Task. Our task is to use deep learning techniques to build a question answering system for the Stanford Question Answering Dataset (SQuAD 2.0). Our model is given a paragraph, and a question about that paragraph, as input. Our model then attempts to output the correct answer to the corresponding question.

Data. We will use the SQuAD 2.0 dataset. As specified in the default final project overview, it contains 129,941 training examples (context, question, answer), 6,078 development examples (context, question), and 5,915 test examples (context, question). Each dev/test question contains three possible answers in order to make evaluation more forgiving. We plan on using the provided starter code for preprocessing the data.

Methods. We plan on implementing the ALBERT model as described in the research paper summary. Once the model is implemented in PyTorch, much of our variations involve ensembling or linguistic knowledge. For ensembling, we expect to implement 5-10 ALBERTs and output the n-best predictions made by all the models. Next, for each n-best prediction, we sum the probability across each model and output the prediction with the highest total probability as our answer. Although our ensemble might contain multiple ALBERTs and potentially a BiDAF, the ALBERT models themselves will be diverse. The ALBERT research paper claims that Dropout negatively affects MLM accuracy, so the default ALBERT model removes Dropout during training. Because there is limited empirical and theoretical evidence that Dropout hurts transformer-based models, we will apply Dropout to a couple of the ALBERT models, as well as various hyperparameter configurations for learning rate, training steps, warmup steps, and maximum sequence length. While ensembling is not an original idea, re-adding Dropout to ALBERT within an ensembling model along with different hyperparameter choices will be unique. As for linguistic knowledge boosting, we will perform post-processing during prediction time to $P(i, j) = softmax(start\_logit(i) + end\_logit(j))$, which represents the probability of a text span $Text(i, j)$ answering the question correctly. Specifically, we will apply a “booster" based on the question type. For example, a question beginning with “Where" is likely to have an answer beginning with a word such as “at", “on", “under", “in", etc. Therefore, if $Text(i)$ is one of these words, we might apply a boost of 0.1 or 0.2 to $P(i, j)$. While other linguistic knowledge methods have been tried, our linguistic knowledge booster will be original for the SQuAD 2.0 question answering task.

Baselines. We will use the BiDAF model as our first baseline to ensure the code is running properly. Then we will implement the default ALBERT-large model as our second baseline. The BiDAF model is downloaded in the starter code, and we will use the code at `https://github.com/huggingface/transformers` to re-implement the ALBERT system with PyTorch. Much of the baseline work will be adapting ALBERT for question answering (in-depth walkthrough in the BERT paper [2]) and adapting the weights for use with PyTorch.

Evaluation. We are using the two standard SQuAD performance metrics: Exact Match (EM) and F1 score. Our final evaluation will take the maximum F1 and EM scores across the three human-provided answers for each question. The EM and F1 scores are then averaged across the entire evaluation dataset to get the final scores.

References

[1] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv e-prints*, page arXiv:1909.11942, Sep 2019.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, page arXiv:1810.04805, Oct 2018.

[3] Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.