Fill the GAP: Exploiting BERT for Pronoun Resolution

Kai-Chou Yang[1], **Timothy Niven**, **Tzu-Hsuan Chou**, and **Hung-Yu Kao**

Intelligent Knowledge Management Lab
National Cheng Kung University
Tainan, Taiwan
zake7749@gmail.com[1]

Abstract

In this paper, we describe our entry in the gendered pronoun resolution competition which achieved fourth place without data augmentation. Our method is an ensemble system of BERTs which resolves co-reference in an interaction space. We report four insights from our work: BERT's representations involve significant redundancy; modeling interaction effects similar to natural language inference models is useful for this task; there is an optimal BERT layer to extract representations for pronoun resolution; and the difference between the attention weights from the pronoun to the candidate entities was highly correlated with the correct label, with interesting implications for future work.

1 Introduction

The Gendered Ambiguous Pronouns (GAP) dataset[1] (Webster et al., 2018) addresses gender bias by providing a dataset balanced over male and female pronouns. The task is made challenging by long paragraph lengths of multiple sentences with a variety of named entities. The text comes from the encyclopedia genre which is more formal and contains numerous technical terms. Furthermore, world knowledge is indispensable to this task. An example is given in Figure 1 where the pronoun is highlighted in green and the entities are highlighted in blue. To know that **She** refers to **Christine** rather than **Elsie Tanne**, a model requires knowing that "never been mentioned again" is a result of having died.

Due to the small size of the dataset (Table 1), our solution was mainly based on transfer learning. Specifically, we used representations of the pronoun and entities from an optimal frozen BERT layer (Devlin et al., 2018) as inputs to a novel encoder architecture (Figure 2), whose results were then ensembled (Maclin and Opitz, 2011) over various BERT models (base and large, cased and uncased) using shallow neural networks.

Christine sent a telegram to congratulate Elsie and Steve Tanner on their wedding day in 1967. In 1984, Elsie Tanner informed Ken Barlow that Christine had died of liver failure after becoming an alcoholic in the late 1970s. She has never been mentioned again.

Figure 1: An example in GAP dataset. To refer "she" to "Christine", a model has to connect "never been mentioned again" with "had died of liver failure", which requires world knowledge.

Our result achieved fourth place in the Kaggle shared-task competition[2]. The competition is composed of two stages. In the first stage, the the development set of GAP which is used for evaluation and is entirely public to help competitors search for model architectures. In the second stage, a large and unpublished dataset was used to test generalization ability as well as prevent label probing.

Our model makes the following contributions to this task. We propose a multi-head natural language inference (NLI) encoder which resolves co-reference though heuristic interaction and efficiently addresses the redundancy in BERT by applying dropout to inputs directly. With layer-by-layer exploration, we extract the task-specific features from the optimal layer of BERT for coreference resolution where we observe pronouns strongly attend to the corresponding candidate entities.

[1] https://github.com/google-research-datasets/gap-coreference

[2] https://www.kaggle.com/c/gendered-pronoun-resolution/over

Proceedings of the 1st Workshop on Gender Bias in Natural Language Processing, pages 102–106
Florence, Italy, August 2, 2019.

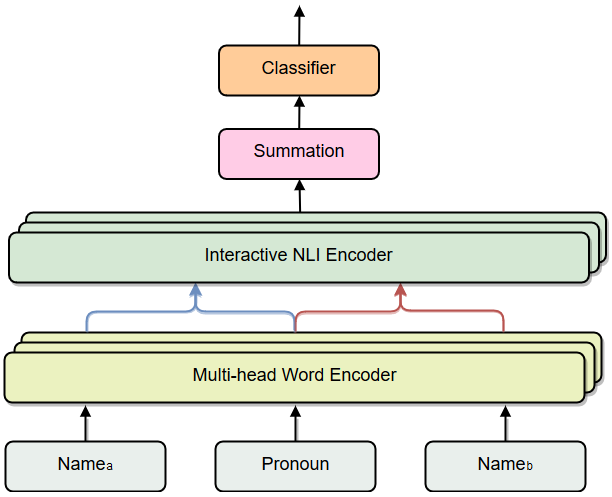

Figure 2: Our multi-head NLI encoder model. The inputs are representations of the pronoun and candidate entities from frozen BERT models.

Dataset	Size	Label Distribution		
		A	B	Neither
Validation	454	0.412	0.452	0.136
Development	2,000	0.437	0.463	0.100
Testing	2,000	0.459	0.428	0.113

Table 1: GAP dataset sizes and label distributions.

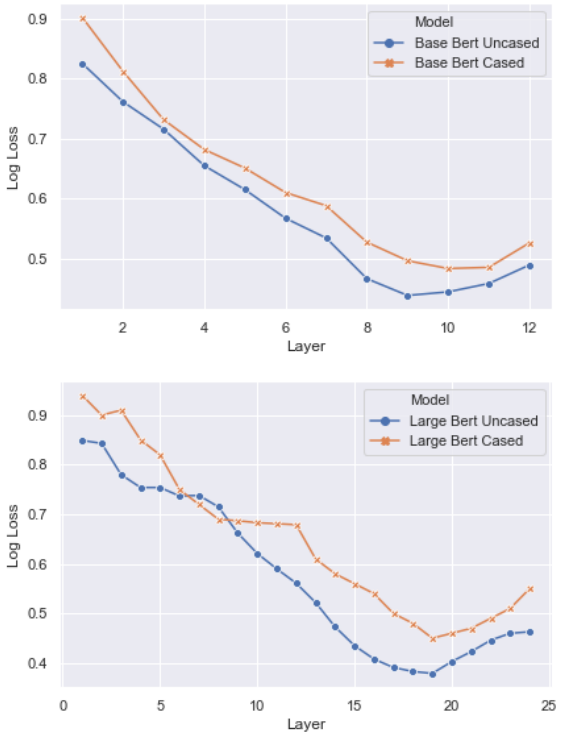

Figure 3: Performance of representations from different BERT layers. Log loss is calculated with the first stage evaluation set.

2 Methodology

2.1 Overview

We use the GAP dataset to train and evaluate our model. For stage 1, following the evaluation settings of Kaggle, our models were trained with the test set, validation set and evaluated on the development set. For stage 2, the models were trained with the whole GAP dataset. The number of samples and distribution of labels in each dataset is shown in Table 1.

Given a query $(entity_a, entity_b, pronoun)$, we obtain deep contextual representations from the optimal layer of BERT for the pronoun and each entity. Where the entities are composed of multiple word pieces, we take the mean of those vectors.

The relations between query words were modeled by two base neural architectures. The first architecture concatenates the contextual representations and aggregates the features with a shallow MLP, which turns out to be simple yet efficient. The second architecture is based on natural language inference. It projects the contextual representations into several low-dimensional subspaces to extract task-specific features which are then passed to an interaction layer and Siamese encoders. We chose the second method as our main model because it is more structural and interpretable. Each base model is trained on frozen BERT Base and Large, and from both cased and uncased versions. The final prediction was an ensemble over all types of base models using a densely connected neural network.

2.2 Optimal BERT Layer

Unlike the common practice of taking the last hidden layer of BERT, we searched for a task-specific layer to obtain the most relevant contextualized representations, which yielded significant improvements. As pointed out by Peters et al. (2018), the hidden layers in language models encode different linguistic knowledge. We therefore conducted a layer by layer analysis to find the best features for this task. As shown in Figure 3, not all BERT layers performed equally well. A similar pattern is observed in both BERT Base and Large, where performance increases until around three quarters of the depth, before becoming worse. The optimal layers turn out to be 8 and 19 for BERT Base and Large, respectively.

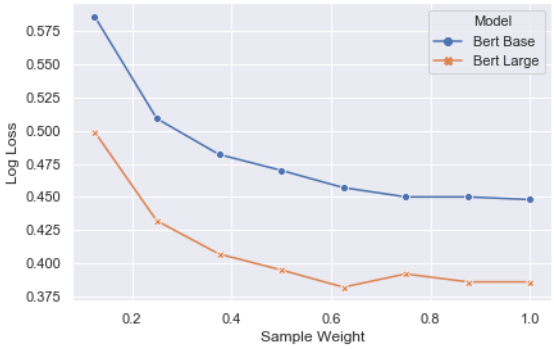

Figure 4: Redundancy in BERT Base and Large. Log loss is calculated with the first stage evaluation set. We can still achieve close to the best log loss in each case by taking a subset of the BERT vectors. This has a regularization benefit overall.

2.3 Redundancy in BERT

Manual analysis of the progression of attention distributions through each layer of BERT indicated the potential for a lot of redundancy in the information attended to, and therefore encoded. We empirically tested this idea by randomly sampling a subset of features in the BERT vectors (held constant through training) and comparing log loss to the sampling rate. Performance degradation halts well before the entire BERT vector is included. This indicates information necessary for this task is present in about 70% to 80% of the BERT vectors (taken from the optimal layer).

Based on this observation, we leverage the redundancy by adding dropout(Srivastava et al., 2014) directly to BERT features, which can be considered a kind of model boosting (Breiman, 1998), similar to training several prototypes with subsets that are randomly sampled from the BERT vector, and letting the neural network figure out the best ensemble architecture through back propagation.

The idea of sampling subsets and then training several models is quite similar to random forests (Breiman, 2001). However, unlike the general situation in random forests where the performance of base models degrade when using less data, our base models can still achieve strong performance because of the redundancy in BERT.

2.4 Word Encoder

Because of the small size of the dataset and the redundancy in the BERT vectors, we chose to project the BERT representations into several lower-dimensional spaces using multiple affine layers with SELU activation (Klambauer et al., 2017), rather than using the whole BERT vector directly. For a k-dimensional word embedding w, the word encoder of the head h encodes it as a n-dimensional vector, which is described as:

$$x_h = \text{SELU}(\mathbf{W_{e_h}} w + b_{e_h}) \quad (1)$$

where $\mathbf{W_{e_h}} \in \mathbb{R}^{n \times k}$ is shared for the pronoun and entities. Our word encoder was also inspired by the multi-head attention (Vaswani et al., 2017), an essential component in BERT, which projects hidden representations into multiple sub-spaces where it can infer the relations between query and key efficiently. We attempted to use more complicated architectures, such as independent transformations for pronouns and entities, or deeper highway transformations (Srivastava et al., 2015), which all resulted in overfitting.

2.5 Modeling Interactions

This task can be considered a sub-task of binary answer selection. We found success modeling interactions between the representations of each entity and the pronoun. We use an established technique from successful natural language inference models (Mou et al., 2015) to model the interaction of encoded results from every head. For the head h, the interaction between encoded entities a_h, b_h and the pronoun p_h is modeled as:

$$i_{ah} = [[a_h; p_h]; a_h - p_h; a_h * p_h] \quad (2)$$
$$i_{bh} = [[b_h; p_h]; b_h - p_h; b_h * p_h] \quad (3)$$

where $i_{ha,hb} \in \mathbb{R}^{4n}$ and $[.;.]$ is the concatenation operator. The interaction vectors are then aggregated to m-dimensional vectors by Siamese encoders which share parameters for (a_h, p_h) and (b_h, p_h), but not shared to different heads:

$$e_{ah} = \text{SELU}(\mathbf{W_{n_h}} i_{ah} + b_{n_h}) \quad (4)$$
$$e_{bh} = \text{SELU}(\mathbf{W_{n_h}} i_{bh} + b_{n_h}) \quad (5)$$

where $W_{n_h} \in \mathbb{R}^{m \times 4n}$. We then sum the results from each Siamese encoder to gather the evidence from all heads:

$$e_a = \sum_h e_{ah} \; ; \; e_b = \sum_h e_{bh} \quad (6)$$

2.6 Handcrafted Features

We also manually create features from parse trees which were generated by spacy,(Honnibal and

Model	Stage 1	Stage 2
Handcrafted features baseline	0.60	-
BERT (B)	0.50	-
BERT (B) + drop.	0.45	0.38
BERT (B) + drop. + inter.	0.43	-
BERT (B) + drop. + inter. + proj.	0.39	0.32
BERT (B) + all*	0.38	0.32
BERT (L) + drop.	0.39	0.31
BERT (L) + drop. + inter. + proj.	0.32	0.24
BERT (L) + all*	0.31	0.24
BERT (B) and BERT (L) ensemble	0.30	0.18

Table 2: The performance gain of each component for both Bert Base and Bert Large, where **all** denotes using input dropout, interaction layers, projection, and handcrafted features at the same time. Performance is log loss from the stage 1 and stage 2 testing data.

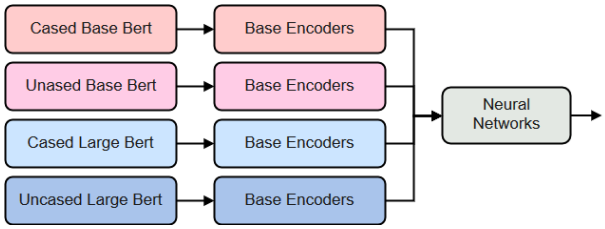

Figure 5: Overview of our ensemble.

Montani, 2017) and from heuristics such as the number of words between the candidate and pronoun. We normalized the hand-crafted features into the range $[0, 1]$ and aggregated them using a feature encoder which consists of an affine layer with SELU activation. To combine the handcrafted features with our encoder architecture, we concatenate the outputs of feature encoder with e_a and e_b.

2.7 Putting it Together

Table 2 shows the performance gains of implementing input dropout, projection, interactions and handcrafted features. In the case **all**, handcrafted features are included. We saw a steady improvement in performance implementing these together that carried over from BERT Base to BERT Large on the both stage 1 and stage 2 evaluation data.

2.8 Ensemble

Figure 5 shows the overview of our ensemble architecture. The ensemble was composed of models whose inputs were from different types of BERTs and handcrafted features. We then aggregated the predictions of the base models with a five-layer densely-connected feedforward network.

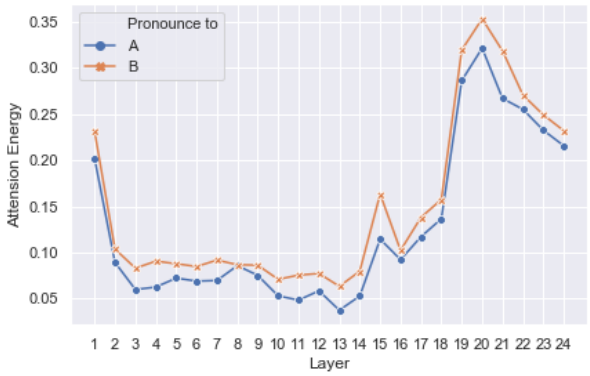

Figure 6: Average attention energy from the pronoun to the entity A and B as a function of layer. Due to imbalance in the dataset, B is more likely to be the correct answer, which explains BERT's preference for B.

2.9 Experiment Setting

Our models were built with Keras. The input dropout rates were 0.6 and 0.7 for BERT Base and Large, respectively. For the concatenation based model, the classifier was composed of a single hidden layer with size 37 following a batch normalization layer and a dropout layer with rate 0.6. For the multi-head NLI encoder, the number of heads was 6 and the dimension of the down-projected vector space was 37. The interactive encoder as composed of a hidden layer with size 37 following SELU activation. To summarize the output from each NLI encoders, we used either a concatenation or summation operation following a dropout layer with rates 0.8, 0.85 respectively. The classifier of the multi-head NLI encoder was exactly the same as the concatenation based encoder.

For training, we validated our models with 7-fold cross-validation and early-stopping on cross-entropy with patience 20. The batch size was 32 and the optimizer was Adam (Kingma and Ba, 2014) with initial learning rate $1e^{-3}$ for all models. We regularized the output layer with 0.1 L2 penalty. The overall training time was about 8 hours for stage 1. For other detailed settings and hyper-parameters please refer to our public code repository.[3]

3 Conclusion

We have demonstrated that BERT has an optimal layer for this task. We also showed that BERT's representations contain redundant infor-

[3] https://github.com/zake7749/Fill-the-GAP

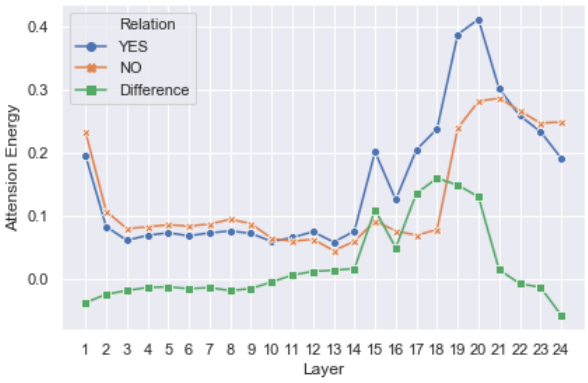

Figure 7: Average attention energy from the pronoun to the correct (YES) and incorrect entity (NO), and the difference between them. These attention distributions provide a signal may be useful for further improvements to our solution.

mation, and that dropout can be used to overcome this problem. Projecting to lower dimensions with multiple heads also allowed us to consider multiple perspectives on this information in more tractable space. Considering interactions between these perspectives also proved beneficial for this task. However, manual error analysis still revealed a large number of world knowledge cases, a major limitation of our solution.

4 Future Work

Post-competition analysis revealed that the difference between attention weights from the target pronoun to the candidate entities in the optimal layer was found to be highly predictive of the correct label. In Figure 6 we can see the overall energy of attention from the pronoun to the candidate entities' peaks. Notice that due to imbalance in the dataset, B is more likely to be the correct answer, which explains BERT's preference for B. Figure 7 shows the average difference in attention energy from the pronoun to the entity that is referred to, or not referred to, and the difference. There are significant gaps between the correct and incorrect candidates in layers 17 to layers 20. The pattern of attention energies is consistent as the observations in section 2.2, which indicates that instead of using the attended vectors, the energies in the attention process can also be efficient and highly-interpretable features for correference resolution.

In future work, we intend to add features from BERT's attention layers to see if we can improve our performance. Furthermore, this discovery could lead to a more general pronoun resolution technique based on BERT that doesn't require candidate entity labeling. We would also like to investigate using this signal for unsupervised and semi-supervised pronoun resolution.

References

Leo Breiman. 1998. Arcing classifier (with discussion and a rejoinder by the author). *Ann. Statist.*, 26(3):801–849.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

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

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. Self-normalizing neural networks. *CoRR*, abs/1706.02515.

Richard Maclin and David W. Opitz. 2011. Popular ensemble methods: An empirical study. *CoRR*, abs/1106.0257.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2015. Recognizing entailment and contradiction by tree-based convolution. *CoRR*, abs/1512.08422.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *CoRR*, abs/1505.00387.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the gap: A balanced corpus of gendered ambiguous pronouns. In *Transactions of the ACL*, page to appear.