Project Report: Gendered Pronoun Coreference Resolution

Justin Payan, Dmitrii Petrov, Subendhu Rongali, Derek Tam
College of Information and Computer Science
University of Massachussets, Amherst, USA
{jpayan, dmpetrov, srongali, dptam}@cs.umass.edu

1 Problem statement

Coreference resolution is the task of identifying all mentions of entities and events in text and placing them into equivalence classes. It is a challenging and unsolved problem which is useful for numerous downstream tasks such as question answering, document summarization, and information retrieval.

The goal of our project was two-fold. First, we wanted to obtain theoretical and practical knowledge of the state-of-the-art coreference resolution approaches. Second, we wanted to build our own coreference resolution system which is gender fair. Recently it has been shown (Webster et al., 2018; Zhao et al., 2018; Rudinger et al., 2018) that current state-of-the-art coreference resolution systems are biased by gender, which may impair their applicability to downstream tasks.

The community has only just begun to address gender imbalance in coreference resolution accuracy, leaving significant room for improvement. However, (Webster et al., 2018) found that simple syntax-based baselines perform reasonably well on their newly introduced Gendered Ambiguous Pronouns (GAP) dataset. These syntactically motivated baselines also show improved gender balance. This indicated that neural models that incorporate syntax may hold promise. In addition, previous work found that attention-based models implicitly learn to do anaphora resolution (Vaswani et al., 2017; Voita et al., 2018), which suggests that these approaches could yield positive results in debiased coreference resolution. We explored these methods in our project, in addition to the baselines from GAP and neural baselines. We found that a reasonably simple task specific architecture added to BERT was able to outperform the baselines by a large margin. There is still room for improvement, especially in cases involving complicated narrative roles, domain-specific knowledge, or complicated syntactic constructions.

2 Accomplished goals

1. ~~Run syntactic baselines~~
2. ~~Submit initial model to Kaggle~~
3. ~~Build and test various models for task (3 weeks)~~
4. ~~Write progress report~~
5. *Select and submit the final model for the Stage 1 of the Kaggle competition*: We didn't do this because we couldn't get our final model in time for the Stage 1 of the submission.
6. *Submit description of model to Kaggle*: We didn't do this because we couldn't submit our final model earlier in time for the Kaggle Competition.
7. ~~Analyze the output of the model, do an error analysis~~
8. ~~Work on final report and presentation~~

3 Related work

Coreference resolution is a hard problem well recognized in the natural language processing community (Pradhan et al., 2012; Doddington et al., 2004). Traditionally, it is seen as a machine learning problem where one clusters all coreferring mentions in a document or classifies pairs of mentions as coreferring or not (Ng and Cardie, 2002; McCarthy and Lehnert, 1995; Soon et al., 2001; Clark, 2015). One of the main drawbacks of these models is that coreference resolution often requires world knowledge, which can be hard to define and incorporate.

Recent advances in deep learning have led to novel approaches for coreference resolution. For

example, (Clark, 2016) use a feed forward neural network to encode pairs of mentions and pairs of clusters before scoring the encodings with another neural network. (Lee et al., 2017) propose an end to end neural approach, assigning each span in the text an antecedent which then implicitly defines a final clustering. Notably, span representations are learned using an LSTM over the span and an attention method to determine the head of the span. While improving coreference resolution these models still suffer from lack of world knowledge.

Recent work has shown that pretrained language models in combination with attention-based models can be useful for more efficient utilization of contextual information. For example, (Peters et al., 2018) pretrain a deep stack of bidirectional LSTM's on the language modeling task and use a weighted sum of the hidden states at each layer as the pretrained embedding. Lee (2018) incorporates these pretrained embeddings and achieves new state of the art on the CoNLL 2012 coreference task (Pradhan et al., 2012).

Although coreference resolution is far from being solved quantitatively, there is an important qualitative issue associated with the aforementioned approaches. Recently, Rudinger et al. (2018) evaluated three coreference resolution systems and found systematic gender bias in each: for many occupations, systems strongly prefer to resolve pronouns of one gender over another (see Figure 1). Authors argue that these systems overgeneralize the attribute of gender, leading them to make errors that humans do not make. Zhao et al. (2018) made similar observations as well.

This problem is closely related to the general line of research concerned with removal of gender bias from natural language processing models. For example, it has been shown that standard word analogies in the Word2Vec or GloVe models are heavily gendered, and that gender appears as a subspace in word embedding space implying that the gender subspace can be subtracted from word embeddings as a first step towards debiasing (Bolukbasi et al., 2016).

4 Dataset

The GAP dataset, released by Google AI Language, is introduced in (Webster et al., 2018) and

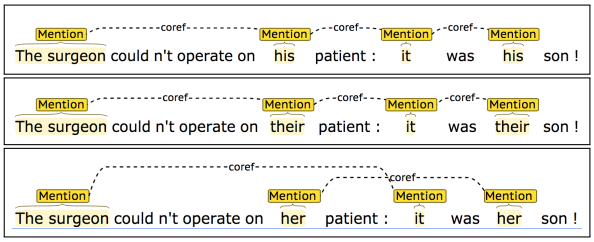


Figure 1: An example of gender biased coreference resolution by (Rudinger et al., 2018). Stanford CoreNLP rule-based coreference system resolves a male and neutral pronoun as coreferent with "The surgeon," but does not for the corresponding female pronoun.

is freely available online[1]. It consists of 8,908 labeled pairs of (ambiguous pronoun, antecedent name) sampled from Wikipedia and divided into train (4,000 pairs), test (4,000 pairs) and validation (908 pairs) sets. The Kaggle competition includes an additional hidden test set, on which submissions were evaluated during a second round. Pronoun gender balance in the dataset is 1:1.

Each example includes a single ambiguous pronoun and two options for its referent. These pronouns and referents are presented within a multi-sentence context, which is a snippet of text from a single Wikipedia page. The authors of the dataset intended to focus on named entities as potential referents, rather than nominals or definite descriptions as in other existing datasets (Zhao et al., 2018). All pronouns are singular, gendered, and non-reflexive. In addition, the two options for the referent always have different head tokens. The task is to select one of the referents, or indicate that neither referent is acceptable. In each example, both referents are of the same gender, meaning that the task cannot be solved by simply matching the referent gender to the pronoun gender. The 1:1 gender balance and the homogeneity of gender within each example imply that gender information alone cannot significantly affect performance on this task. This point is crucial, as it implies that much of the gender bias typically inherent in pronoun coreference resolution datasets simply is not present in this dataset.

Some example contexts from the publicly available test set are illustrated in Table 1. These examples were specifically selected to illustrate some of the reasons why coreference resolution is challenging. Test-42 is difficult because the original

[1]Publicly available at github.com/google-research-datasets/gap-coreference

Ex. #	Source Article	Referent	Text
Test-42	Bossier Parish, Louisiana	Neither	[...] Jerry Miculek, American professional speed and competition shooter known for **his** 20 world records; resides in Princeton **George Nattin**, mayor of Bossier City, 1961-1973 **Ford E. Stinson**, Jr., retiring chief judge of the Louisiana 26th Judicial District Court [...]
Test-66	Circuit de Charade	**Helmut Marko**	[...] a stone thrown from **Emerson Fittipaldi**'s Lotus penetrated the helmet visor of **Helmut Marko**, blinding **him** in the left eye and ending his racing career.
Test-1263	Maud Barger-Wallach	**Wood**	[...] when Sidney Wood of the U.S. won the Wimbledon Gentlemen's Singles over **Frank Shields** of the U.S. in the final, in a walkover because Shields was injured, **Wood** gave **his** trophy to Barger-Wallach [...]

Table 1: Examples from the publicly available test set.

context is in a list format. This fact is difficult even for humans to ascertain, since the formatting is not preserved in the GAP dataset. Test-66 and test-1263 are challenging because they may require commonsense reasoning. In test-66, the coreference decision requires knowledge that the person on the receiving end of the stone is likely to be blinded, rather than the subject throwing the stone. In test-1263, proper coreference requires knowledge that the winner of a tennis match receives a trophy. This example additionally shows that even if the title of the article appears in the context, it is not always the correct referent.

4.1 Data preprocessing

Our final BERT model combines syntactic rules with an end-to-end neural architecture. To do this, we had to obtain syntactic information about each of the tokens in the reference text, including the pronoun and the candidate coreferents.

Based on the rules for the syntactic baselines in the original GAP paper (Webster et al., 2018), we decided to look at the following syntactic information for each of the tokens. We list the standard NLP model pipeline used to obtain each tag in parentheses.

- The part-of-speech tag (POS tagging)
- The dependency tag (Dependency Parsing)
- The entity type tag (Named Entity Recognition)

To do this efficiently in the pre-processing step, we used an existing NLP pipeline for the English language provided by spaCy[2]. spaCy has the fastest syntactic parser that's freely available, and its accuracy is within 1% of the state-of-the-art models (Choi et al., 2015). The new version claims even better performance.

We processed each of the reference texts in the train, validation and test datasets by sending them through the *en-core-web-sm* model from spaCy. We then extracted the required part-of-speech, dependency and entity type tags from the tokens and augmented our datasets that we sent to the neural model.

4.2 Data Errors

Despite how small the dataset was, there were still quite a bit of errors in the dataset. There were 74 incorrect labels in train and 85 incorrect labels in dev and test, according to another competitor in the Kaggle competition[3]. We manually identi-

[2] spacy.io

[3] Provided in https://storage.googleapis.com/kaggle-forum-message-attachments/

Ex. #	Annotated Referent	Correct Referent	Text
Test-15	**Maria**	**Gracia**	**Maria**'s mother, **Gracia**, wanted **her** daughter to catch this rich man at all costs and convinced her that pregnancy would assure this.
Test-36	**Cromwell**	**Lim Goh Tong**	However, **Cromwell** has successfully gained financial backing for the tribe's casino development effort from Malaysian billionaire **Lim Goh Tong** and **his** Kien Huat Realty arm of the Genting Group [...]
Test-221	**George W. Bush**	**George W. Bush**	In his speech at the 2002 signing of the Born-Alive Infants Protection Act President **George W. Bush** mentioned **Jessen**, acknowledging her presence and extending **his** appreciation.

Table 2: Examples of annotation errors in the test set.

fied a small handful of examples with the incorrect coreferent entity annotated, and examples where the genders of the candidate coreferents did not match. A handful of such examples are included in Table 2. Note that a handful of examples such as Test-221 had the correct referent labeled, but the gender of the two candidates did not match.

5 Baselines - Syntax based approaches

We use a random baseline, as well as three syntactic baselines described in (Webster et al., 2018).

The original paper that introduced the GAP dataset (Webster et al., 2018) mentions a series of syntax-based approaches to solve the task of pronoun resolution. These approaches are based on heuristics proposed from syntactic information in the given text. The required syntactic information is extracted through existing NLP tools - a part-of-speech tagger and a dependency parser. The authors originally use these tools from the Google Cloud Platform, but for our implementation we use an existing kaggle kernel (Attree, 2019) that uses tools from AllenNLP, SpaCy, and Stanford CoreNLP.

We use three main baseline models that are described below.

- Token distance - In this model, the text is tokenized and we simply pick the candidate with the smaller token distance to the pronoun. So this model always chooses from candidate A or candidate B, never predicting the option - neither. On its own, this model performs similar to a random model, emphasizing the difficulty of the task in the given dataset.
- Syntactic distance - This model first constructs dependency trees for the text. It then chooses the candidate that has the shortest path in the tree from the given pronoun. A tie is broken by backing off to token distance.
- Parallelism - In this model, we pick the candidate whose syntactic dependency relation (subject or dependent object) matches with that of the pronoun. We back off to syntactic distance and then token distance on ties.

Note that these are simply heuristics to choose between candidates. So we don't have any associated probability scores to calculate the log loss for these models. We report the F1 scores and bias factors for these models which are in-line with the scores reported in (Webster et al., 2018) in Table 3.

6 Our approach

Our methods can be split in two groups: without BERT and with BERT. During our experiments we found that BERT representations are extremely useful for this task, so we mainly focus on models with BERT.

We use BERT in the following way: we insert [UNK] tokens before A, B and the pronoun.

503094/12752/corrections.csv located at https://www.kaggle.com/c/gendered-pronoun-resolution/discussion/81331#latest-503495

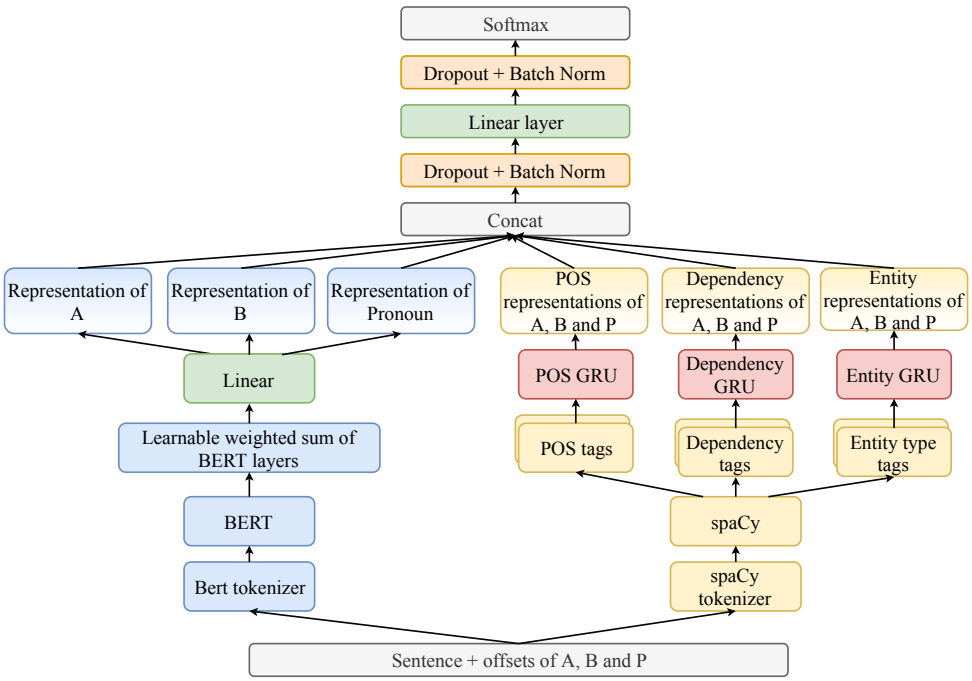


Figure 2: Our final model: We use a *BERT* to embed words. We use *spaCy* to obtain the required syntactic tags. The final classification layer is an *MLP* that consists of multiple batch-norm layers and a linear layer.

Below is an example of a sentence modified for this type of input to BERT:

[CLS][UNK] Kathleen first appears when [UNK] Theresa visits [UNK] her in a prison in London. [SEP]

After running BERT-large, we extract three representations of these three [UNK] tokens and concatenate them into one vector of size 3072 (3x1024). On top of this vector, we add different models described later in this section.

Table 3 shows the F1, log loss, and bias results for this experiment. Note that F1 is only over the Male and Female classes while log loss is over Male, Female, and Neither class.

We used the following libraries and implementations: pretrained Bert from HuggingFace, Spacy, PyTorch, LGBM, and MatPlotLib. We ran on local computers, colab, and servers. We use a kaggle kernel to run the baselines, but we did not include that code in our submission.

The models are in the codebase as follows: all the trained models on BERT-base are in gap/src/models while all the trained models on BERT-large are in the notebooks.

6.1 LSTM-Dot

LSTM-Dot consists of running an LSTM over the words in a sentence and then taking the dot product of the pronoun hidden state with the hidden state of each possible coreferent. The last hidden state of the sentence is used to represent 'Neither' coreference, with cross entropy loss. We did a grid search over embedding dimension: [128, 256] and RNN hidden dimension: [128, 256], taking the highest scoring model on the dev set and evaluating it on the test set.

6.2 BERT + LSTM-Dot

This approach is the exact same as LSTM-Dot, but the embeddings are the last layer of the Transformer in the BERT-base model rather than being learned as in LSTM-Dot. We did a grid search over RNN hidden dimension: [128, 256], taking the highest scoring model on the dev set and evaluating it on the test set.

6.3 Bert + LSTM-Dot + Syntatic Information

This approach consists of concatenating part-of-speech embeddings and dependency tag embeddings to the text embeddings before running it through LSTM and then following the same procedure as Bert + LSTM-Dot.

	F1	Log Loss	Feminine F1	Masculine F1	Bias
Syntactic Baselines					
Random	.47	-	.46	.47	.98
Token Distance	.49	-	.47	.51	.92
Syntactic Distance	.67	-	.67	.67	**1.0**
Parallelism	.69	-	.68	.69	.99
Our models (Neural Networks)					
LSTM-Dot	.58	1.34	.58	.60	.97
Our models (Bert Base)					
Bert + LSTM-Dot	.77	1.32	.77	.78	.99
Bert + LSTM-Dot + Syntatic Information	.77	1.78	.77	.77	.99
Bert + MLP	.75	1.32	.75	.75	**1.0**
Our models (Bert Big)					
Bert + GBDT	.71	0.66	.68	.72	.94
Weighted Bert + MLP	.85	0.55	.83	.86	.97
Weighted Bert + MLP + Syntactic Information	**.87**	**0.45**	**.85**	**.88**	.96

Table 3: Log Loss, Total F1, Feminine F1, Masculine F1, and Bias(F/M) on the GAP dataset for all our models so far

6.4 BERT + MLP

This approach consists of running a feed forward neural network over the hidden states of the pronoun and possible coreference states from the last layer of the Transformer in the BERT-base model. We did a grid search over the output dimension: [64, 128], taking the highest scoring model on the dev set and evaluating it on the test set.

6.5 BERT + GBDT (Gradient Boosted Decision Trees)

On top of these vectors we run gradient boosted decision trees using LightGBM library. We use very shallow trees of depth 3 to avoid overfitting. We also bag and select a sub sample of features for each new tree.

6.6 Weighted Bert + MLP

We applied a heavily regularized linear layer on top of concatenated BERT representations. We use a dropout of 0.5 for BERT representations, and a dropout of 0.5 after the first hidden layer. This type of approach yielded the models with the best performance.

Due to memory issues we could not fine-tune the whole BERT. However, we fine-tuned the last few layers of BERT. Fine-tuning did not improve performance of our model. Instead, we tried to improve BERT representations through a learned weighted sum of all 24 BERT layers in the style of ELMo (Peters et al., 2018). The post-training weights for each BERT layer are illustrated in Figure 3.

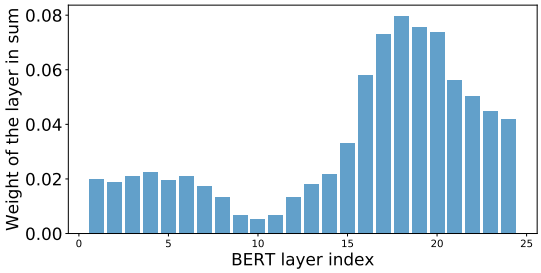
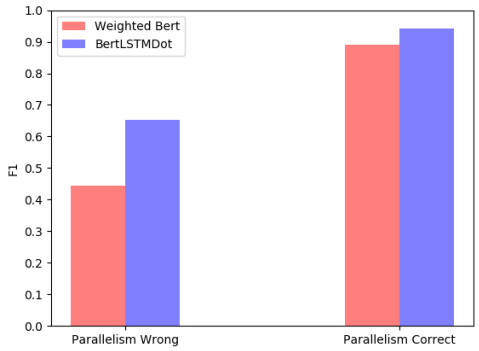


Figure 3: Learned BERT layer weights. In the beginning of the training we initialized all weights uniformly and did sum of the softmax of these weights during training. Figure represents weights in the end of the training.

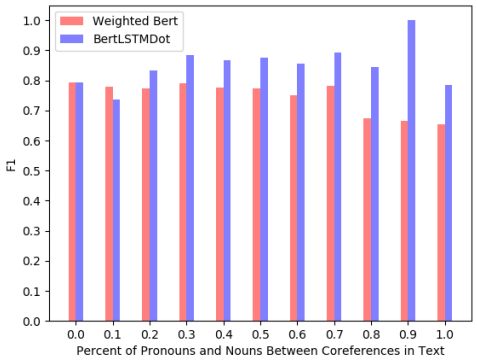
6.7 Weighted Bert + MLP + Syntatic Information

In this model, we incorporate syntactic information from the spaCy parser to the Weighted Bert + MLP model described in the previous subsection. The architecture of this model can be seen in Figure 2.

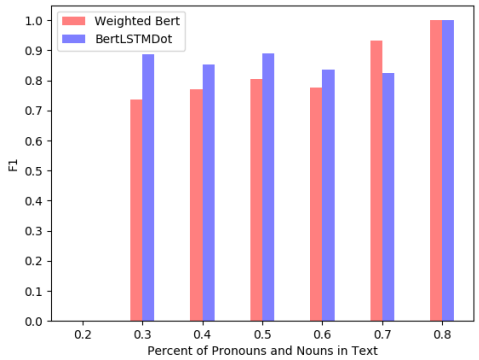
We first extract the part-of-speech, entity type, and dependency tags for each word in the text. We then pass each set of syntactic features through separate embedding layers, followed by separate bi-directional GRUs. We then extract the hidden state representations for the candidates A and B and the pronoun from each of the GRUs. We then concatenate these hidden states to the concatenated BERT representations. We then pass this vector through the MLP as described in the previous model. We used 30 dimensional embeddings for all the syntactic features and GRU hidden states.



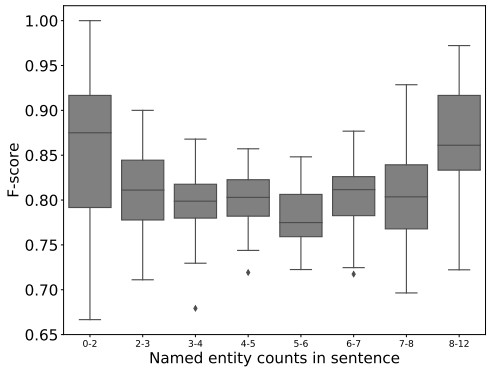
(a) Bar graph of F1 for Weighted Bert and BertLSTMDot for examples in which parallelism was wrong and for examples in which parallelism was correct.



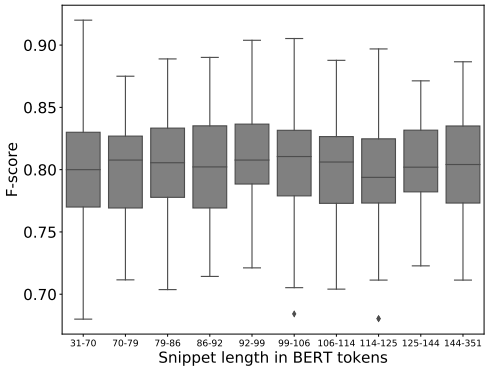
(b) Bar graph of F1 for Weighted Bert and BertLSTMDot for differing percentage of pronouns and nouns in between pronoun and true coreferent.



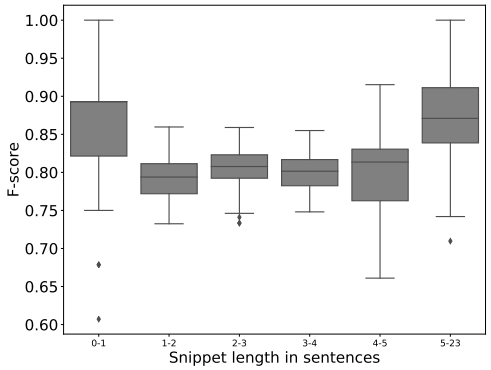
(c) Bar graph of F1 for Weighted Bert and BertLSTMDot for differing percentage of pronouns and nouns in text.



(d) Distribution of F1 depending on number of named entities in sentence (non-unique). Scores were estimated using bootstrapped values of predictions (500 repetitions of 50% test examples).



(e) Distribution of F1 depending on the snippet length in BERT tokens. Scores were estimated using bootstrapped values of predictions (500 repetitions of 50% of test examples).



(f) Distribution of F1 depending on snippet length in sentences (sentence count was done with spaCy). Scores were estimated using bootstrapped values of predictions (500 repetitions of 50% of test examples).

Figure 4: Quantitative analysis

7 Observations and oddities

- *It's hard to beat BERT:* Even simple regularized MLP on top of BERT yields surprisingly strong results. Everything we did provided only minor improvement over this model. It may be due to BERT's power, the weakness of our models, or both.

- *BERT uncased was better than cased:* It was surprising to see that uncased BERT gives better results than cased. One would think that the cased model is better for coreference resolution, which includes a lot of cased named entities. It's hard to pinpoint the nature of this effect, but perhaps the relatively small data size leads to overfitting of the cased model (the uncased model would be affected, but to a lesser extent).

- *Augmentation with [MASK] or [UNK] tokens didn't work:* We tried to augment our data by masking 10-20% of the sentence using MASK or UNK tokens for BERT. It didn't work and lead to overfitting. Augmentation via name replacement could work better, though we did not pursue this approach further.

- *Layer 18 of BERT seems to capture the same information as the last layer for this task:* When we studied learned BERT weights in the weighted average scheme, we found that layer 18 always has the highest weight. We trained our model on outputs of just this layer, and it yielded similar results to the last layer. Other layers performed slightly worse (but we didn't do any statistical analysis on these scores).

- *It's hard to debug CUDA errors:* CUDA errors are hard to trace and pretty cryptic to debug.

8 Error analysis

8.1 Quantitative Analysis

Our model retains performance on examples with correct parallelism predictions and also improves over incorrect parallelism predictions (Figure 4a). This indicates that BERT is able to capture some notion of semantic role labeling, since the parallelism is based off of semantic role labeling. One might expect that as the number of nouns or pronouns between the coreferences or the total number of nouns or pronouns in the text increases, then the performance decreases since the model might get confused with more possible options. Surprisingly though, the performance doesn't change depending on the percentage of text between the two coreferences that are nouns or prounouns (Figure 4b) and the percentage of total text that are nouns or pronouns (Figure 4c).

8.2 Qualitative Analysis

We performed two major qualitative analyses on the Weighted BERT model. We use this model and not the Weighted BERT model with syntactic information because it is simpler and achieves nearly the same performance.

For our first qualitative analysis, we manually evaluated the examples where parallelism held, but predicted the incorrect referent. Of the 99 examples where parallelism existed and led to the incorrect prediction, 69 of them were correctly predicted by Weighted BERT. Manual analysis of these examples shows BERT seems to capture basic world knowledge and higher-order semantics, because many of these examples require such knowledge to get the right answer. Some such examples are reproduced in Table 4.

The Weighted BERT model fixed many examples, but it still failed on others. Of the test examples, we selected all the examples which were annotated as either A or B (not neither). We checked the 175 examples in the test set where BERT predicted either A or B, but chose the wrong answer. We performed manual analysis of the first 50 failure cases of the Weighted BERT model. We found that 15 of these 50 are annotation errors, which BERT generally gets correct and is penalized for. The most prevalent legitimate errors are examples requiring understanding of narrative roles in various frames (10), ability to parse complex syntax or resolve multiple coreference problems at once (7), and domain-specific knowledge (6). Some other challenges are examples that are truly ambiguous (4), examples requiring knowledge of topicality (3), and one example with idiomatic language. There are 4 examples with no discernible challenge.

We document some of these examples in Table 5. Some interesting cases include Test-95, which requires knowledge that someone being

Ex. #	Referent	Challenge	Text
Test-1059	**Abram**	Factoid recall	After **Jones** died in 1903, the house was purchased by **Abram** and Sarah Hewitt; **he** served as Mayor of New York City 1887-88.
Test-1065	**Fred Ordonez**	Narrative roles	**Putnam** invited future member **Fred Ordonez**, from the band Shit Scum, to the show, but **he** failed to turn up.
Test-1844	**Helena**	Long-range semantics	Helena and Alvaro are back together [...] **Pilar** is still very upset with **Helena** because of the relationship **she** has with Alvaro.

Table 4: Examples which the Parallelism baseline predicts incorrectly, but are correctly predicted by Weighted BERT.

hired for a position requires expertise, but also requires the model to incorporate the context that Davidson was the person hired, not Coach Fox. In Test-183, "his" clearly must refer to "King Edward I of England" based on context, but interestingly "his family" refers both to the family of Edward and the family of his brother, the Duke of Lancaster (they have the same family). To say that "his" refers to "Duke of Lancaster" is not semantically incorrect - it would just be highly unlikely following normal speech patterns. In Test-219, the model must essentially perform multi-hop reasoning to perform coreference - as it must know that Ray McKinnon moved to Dundee United, but that Toshney did not. We also note that with examples like Test-232, the model must perform multiple coreference (here, the model should first determine that "his" refers to the first mention of Father Joseph, then determine that the first mention is coreferent with the second mention). This type of example is rather challenging (or impossible), because the model is not trained to do anything other than binary classification given the two candidates and the pronoun. Test-462 indicates a new class of challenging examples which was not noted in the original GAP paper, idiomatic uses of pronouns. To see why this example requires knowledge of English idioms, imagine changing "shrugs her off" to "shrugs her shoulders." This change would result in the referent changing from Alice to Penny. In general, the remaining errors by the Weighted BERT model require highly non-trivial reasoning. Though they provide interesting avenues for future research, it is understandable why the BERT model has trouble with these examples.

9 Contributions of group members

- Justin Payan: writing, qualitative error analysis, annotation error analysis, poster printing.
- Dmitrii Petrov: model development, quantitative error analysis, writing, poster template.
- Subendhu Rongali: syntactic baselines, writing, diagrams, model development.
- Derek Tam: model development, quantitative error analysis, writing.

10 Conclusion

We thought we would have to pay more attention to removing gender bias. However, because the dataset was balanced by gender and most examples (modulo annotation errors) had both candidate referents with the same gender, the gender bias was inherently removed to a large degree in the task formulation. We found that the best model uses a task specific architecture on top of a weighted average of BERT layers, which can be slightly improved upon using more a more complicated task specific architecture with syntactic information. Our results are not particularly surprising, as BERT plus a small task-specific architecture outperforms nearly every model on every dataset at the moment. However, it is interesting that the explicit syntactic features do not significantly improve the model, a fact which seems to indicate that BERT already has learned an implicit representation of syntax. It would be interesting to experiment with datasets that have known gender imbalance, for example training our best model on

a dataset with gender biased training data, then applying the model on the GAP test set. In addition, error analysis of the BERT model indicates that BERT still has trouble with coreference resolution when it requires interpreting narrative roles or using domain-specific knowledge. Either of these two settings would make for interesting new coreference resolution datasets.

Ex. #	Referent	Reason for Failure	Text
Test-87	**Elle Macpherson**	Truly ambiguous	A version of **Julie Madison** appeared in the 1997 film Batman & Robin, played by **Elle Macpherson**. Many of **her** scenes were edited out of the film's final cut, thus she is the only film love interest of Batman's to not have a prominent role.
Test-95	**Davidson**	Narrative roles	Weis is a good friend of Panthers head coach John Fox, and it is believed that his recommendation was instrumental in Fox's decision to hire **Davidson**. With **his** playing and coaching experience at the offensive line and additional coaching with tight ends, most believe this hire meant that Coach **Fox** wanted to keep his focus on a powerful offense based around a strong running game [...]
Test-183	**Edward I of England**	No discernible reason	The Red Rose of Lancaster derives from the gold rose badge of **Edward I of England**. Other members of **his** family used variants of the royal badge, with the king's brother, the **Earl of Lancaster**, using a red rose.
Test-219	**Ray McKinnon**	Domain knowledge	Following **his** move to Scottish Championship rivals Dundee United, former Raith Rovers manager **Ray McKinnon** expressed an interest in bringing **Tosh-ney** to Tannadice.
Test-232	**Father Joseph**	Complex syntax	Father Joseph savors the power [...] He confronts Marie with the announcement of the execution of Cinq-Mars [...] Further, **he** tells her, the Polish ambassador will return soon from a hunt with the **King**, and **Father Joseph** advises Marie to answer him favorably [...]
Test-462	**Alice**	Idiomatic language	**Alice** imagines what is coming and immediately tries to quash the suggestions, but **Penny** shrugs **her** off [...]

Table 5: Examples which Weighted BERT predicts incorrectly.

References

Attree, S. (2019). Reproducing GAP results. https://www.kaggle.com/sattree/2-reproducing-gap-results.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357.

Choi, J. D., Tetreault, J., and Stent, A. (2015). It depends: Dependency parser comparison using a web-based evaluation tool. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 387–396.

Clark, M. (2015). Entity-centric coreference resolution with model stacking. In *Transactions of the ACL*.

Clark, M. (2016). Improving coreference resolution by learning entity-level distributed representations. In *Transactions of the ACL*.

Doddington, G. R., Mitchell, A., Przybocki, M. A., Ramshaw, L. A., Strassel, S. M., and Weischedel, R. M. (2004). The automatic content extraction (ace) program-tasks, data, and evaluation. In *LREC*, volume 2, page 1.

Lee, K., He, L., Lewis, M., and Zettlemoyer, L. (2017). End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*.

Lee, He, Z. (2018). Higher-order coreference resolution with coarse-to-fine inference. In *NAACL-HLT*.

McCarthy, J. F. and Lehnert, W. G. (1995). Using decision trees for coreference resolution. *arXiv preprint cmp-lg/9505043*.

Ng, V. and Cardie, C. (2002). Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 104–111. Association for Computational Linguistics.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. (2012). Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 1–40. Association for Computational Linguistics.

Rudinger, R., Naradowsky, J., Leonard, B., and Van Durme, B. (2018). Gender bias in coreference resolution. *arXiv preprint arXiv:1804.09301*.

Soon, W. M., Ng, H. T., and Lim, D. C. Y. (2001). A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Voita, E., Serdyukov, P., Sennrich, R., and Titov, I. (2018). Context-aware neural machine translation learns anaphora resolution. *arXiv preprint arXiv:1805.10163*.

Webster, K., Recasens, M., Axelrod, V., and Baldridge, J. (2018). Mind the gap: A balanced corpus of gendered ambiguous pronouns. In *Transactions of the ACL*, page to appear.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., and Chang, K.-W. (2018). Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv preprint arXiv:1804.06876*.

A Synthetic Paired Examples

To further test BERT's sensitivity to context and its knowledge of facts about the world, we created seven pairs of coreference problems with subtle changes, which are fully documented in Table 6. The first three are intended to test BERT's ability to notice subtle semantic differences in context which lead to changes in coreference behavior. The fourth tests BERT's knowledge of idiomatic language (it is the same idiomatic language example listed in Table 5 earlier in the paper). The fifth and sixth test BERT's world knowledge. The final example pair tests BERT's knowledge of gender of popular figures. It is clear that BERT has some ability to handle subtle semantic cues in context, but that its knowledge of world knowledge and gender is not very impressive. Based on our previous error analysis and the results in the appendix, we suspect that BERT can handle coreference resolution requiring simple semantic cues and factoid recall. More complicated semantic relationships and detailed semantic recall cause trouble for the model, however.

Referent A (Probability)	Referent B (Probability)	Text
Pilar (19.3%)	**Helena (36.3%)**	*Helena* and *lvaro are back together even though Pilar already knows about this situation and is quite upset that thanks to the insistence of Germ*n is thinking of going with him. **Pilar** is still very upset with **Helena** because of the relationship **she** has with *lvaro.
Pilar (32.8%)	**Helena (22.0%)**	*Pilar* and *lvaro are back together even though Helena already knows about this situation and is quite upset that thanks to the insistence of Germ*n is thinking of going with him. **Pilar** is still very upset with **Helena** because of the relationship she has with *lvaro.
Nicolette (9.0%)	**Barbara (67.4%)**	**Nicolette** helped to comfort **Barbara** while **she** was *sick* and care for her children.
Nicolette (6.6%)	**Barbara (65.4%)**	**Nicolette** helped to comfort **Barbara** while **she** was *free* and care for her children.
Robertson (48.8%)	**Joan Lois Coburn (37.3%)**	**Robertson** was born in Waipukurau in 1953, the daughter of **Joan Lois Coburn** and **her** *husband* Alexander Lawrence Coburn.
Robertson (73.5%)	**Joan Lois Coburn (9.6%)**	**Robertson** was born in Waipukurau in 1953, the daughter of **Joan Lois Coburn** and **her** *father* Alexander Lawrence Coburn.
Alice (35.6%)	**Penny (40.4%)**	**Alice** imagines what is coming and immediately tries to quash the suggestions, but **Penny** shrugs **her** *off* and instructs everyone to write.
Alice (14.3%)	**Penny (63.0%)**	**Alice** imagines what is coming and immediately tries to quash the suggestions, but **Penny** shrugs **her** *shoulders* and instructs everyone to write.
Michael Jackson (26.5%)	**Elvis Presley (26.6%)**	**Michael Jackson** met with **Elvis Presley** at **his** home, *Neverland Ranch*.
Michael Jackson (22.2%)	**Elvis Presley (23.0%)**	**Michael Jackson** met with **Elvis Presley** at **his** home, *Graceland*.
Donald Trump (2.2%)	**Vladimir Putin (65.6%)**	**Donald Trump** greeted **Vladimir Putin** at **his** office in the *White House*.
Donald Trump (2.2%)	**Vladimir Putin (66.5%)**	**Donald Trump** greeted **Vladimir Putin** at **his** office in the *Kremlin*.
Mickey Mouse (2.0%)	**Angela Merkel (77.5%)**	**Mickey Mouse** met with **Angela Merkel**. ***He*** was pleased with the meeting.
Mickey Mouse (2.7%)	**Angela Merkel (77.9%)**	**Mickey Mouse** met with **Angela Merkel**. ***She*** was pleased with the meeting.

Table 6: Synthetic pairs of examples which illustrate BERT's performance in changing contexts. Modified context is italicized.