Gendered Pronoun Resolution

Apoorv Chaudhary
Department of Computer Science
Stanford University
apoorvc2@stanford.edu

Abstract

We implement an ensemble of various neural architectures to solve a popular problem in the Natural Language Processing (NLP) space: gendered pronoun resolution. This gender imabalance generally results from there being more examples and entries of male pronouns versus females. Pronoun resolution falls under the umbrella of coreference resolution, the task of properly resolving these ambiguous pronouns. Architectures used in this project are an Attention Recurrent Neural Network (RNN), a Multilayer Perceptron (MLP) Convolutional Neural Network (CNN), and an ASGD Weight-Dropped LSTM (AWD-LSTM). We observe AWD-LSTM to perform the best out of the three models with a logloss of 0.6897 while the overall ensemble gives us a logloss of 0.6201.

1 Introduction

Coreference resolution is a field which essentially involves finding all expressions that refer to the same entity in a text. While systems for coreference resolution are able to perform well on shared tasks [1], this success does not translate well to downstream tasks such as machine translation [2] and fact extraction [3]. Strong systems are able to properly identify coreference relationships by direct string matching of proper names, but fall short on parts of speech such as pronouns and common noun phrases [4].

Another common issue in the space is an inherent imbalance in the gender of the referring pronouns where datasets contain more male pronouns than female. The Mind the GAP [5] project aims to tackle this problem by creating a balanced corpus of gendered ambiguous pronouns. It is important to resolve this bias as this leads to underrepresented groups being miss-represented in downstream processes [6].

In addition to tackling gender bias, this paper creates various baselines and benchmarks to improve upon. One of the primary objectives of our project was to begin with a recreated baseline of 0.8388 logloss and begin to improve upon that with various neural architectures and strategies. If this bias is able to be tackled with proper methods, the hope is that we can provide an effective strategy for tackling a challenging problem while dealing with bias.

2 Related Work

Prior to GAP much work had been done in the ambiguous pronoun identification space. It is crucial to properly understand prior works which led to the necessity of the GAP dataset towards better solving this pronoun resolution problem in an unbiased manner.

2.1 Ambiguous Pronoun Datasets

One area which relates to pronoun resolution are Winograd Schemas. These are a pair of sentences which differ in only one or two words and contain an ambiguity that is resolved in opposite ways in

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

the two sentences. Winograd Schemas also requires usage of world knowledge and reasoning for resolution [7]. There have been a variety of Winograd Schema challenges and datasets of which the largest dataset contained 3,160 examples created by Zhao et al. [8].

While the Winograd Schema datasets did an excellent job capturing a source of ambiguous pronouns, they did not generalize well and were very carefully curated. In terms of general coreference datasets OntoNotes [8], provided a good collection of simpler, high-frequency coreference examples, but did not have many examples of ambiguous pronouns. In order to solve this, Ghaddar and Langlais released WikiCoref [9], a corpus of 30 articles annotated with coreference.

The examples in the GAP dataset are similar to Winograd Schemas in that they contain two person named entities of the same gender with an ambiguous pronoun which could potentially refer to either. They are different in that they have no reference-flipping word, but still represent a comparable challenge and require similar inferential power. GAP is larger in size than both Winograd Schema datasets as well as WikiCoref and solves the problem of pronoun resolution better than OntoNotes due to its targeted nature.

2.2 Ambiguous Pronoun Modeling

As our dataset is similar in many ways to the Winograd Schemas datasets, it is of value to analyze methods of modeling used to solve that problem. A notable score on a Winograd dataset emerged when Rahman and Ng [10] scored 73.05% on their dataset through the usage of narrative chains, Web-based counts, and selectional preferences. This was improved upon by Peng et al. [11] to a 76.41% by implementing triplets of (subject, verb, object) and (subject/object, verb, verb).

Other challenges which are also relevant to study for modeling inspiration are the Winograd Schema Challenge datasets as well as the Pronoun Disambiguation Problem. Trinh and Le [12] set benchmarks for both of these with scores of 63.7% and 70% respectively. They chose to ensemble of word-level and character-level recurrent language models which heavily serves as inspiration of our approach on this project.

2.3 Machine Learning Biases

One of the biggest differences with this dataset and our approach is that we aim to tackle gender bias which has been prevalent in previous datasets. The OntoNotes test corpus contained approximately 2,000 gendered pronouns of which less than 25% were feminine [8]. The Definite Pronoun Resolution Dataset training data did not do a much better job at tackling this problem as it had 27% of the examples representing the female gender. WikiCoref was the most shocking in this situation as it contained only 12% femine examples. The GAP dataset solves the WikiCoref dataset's issues in a large way as the GAP dataset is taken from Wikipedia articles. Two popular Winograd Schema Challenge datasets also fell victim to this issue boasting only 28% and 33% feminine examples. This was recently tackled via the WinoBias [8] and Winogender schema [13] datasets as they balanced the number of feminine examples. The primary difference between these and the GAP dataset is that these focused on pronoun coreference where the antecedent is a nominal mention while GAP focuses on relations where the antecedent is a named entity. As we think about the future of the coreference problem and general NLP problems, dealing with gender bias is crucial from both an ethical and practical standpoint which is why this project has chosen to work with the GAP dataset.

3 Approach

In this section we discuss and detail the three different architectures we have employed on this problem thus far. The last section will discuss our approach to ensembling. The baseline that we have chosen was an attempt to reproduce the results by Webster et al. [5] in their Mind the Gap paper. Scored on the current evaluation metric, this logloss was 0.8388. This project aims to extend and improve upon this baseline by minimizing the logloss.

3.1 Attention RNN

The first neural network employed is an Attention RNN. This is essentially a modified version of a state of the art model presented in End-to-end Neural Coreference Resolution by Lee et al. [14].

The modification here is that we only need to concentrate on a specific case of the generic reference problem and we only have to choose from two classes (male and female).

We have multiple initial layers which take the initial input and transform it with the appropriate features. Then we feed this into embedding layers which properly embeds the data. This can be seen in the images below [14]:

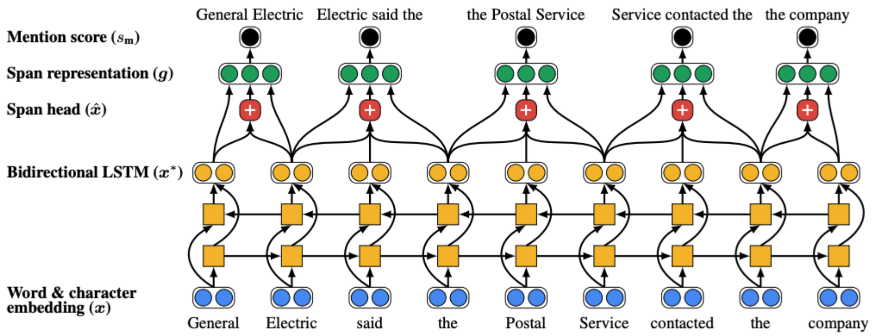

Figure 1: This is the first step of the end-to-end coreference model. This computes embedding representations of spans for scoring potential entity while also pruning low-scoring spans.

The pruned spans are then processed as follows:

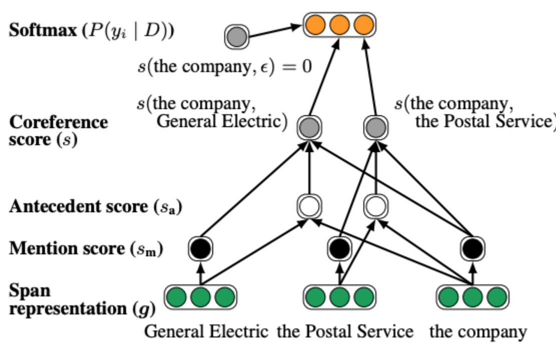

Figure 2: In the second step, antecedent scores are computed from pairs of span representations. The final coreference score of a pair of spans is computed by summing up the mention scores of both spans and their pairwise antecedents.

We then define our bidirectional lstm layer followed by a batch normalization layer. This feeds into a feature selection layer which is followed by flatten layers and attention layers.

We define a class for the Attention operation which also includes a context vector for temporal data. This is based off the work of Yang et al. [15], which also uses a context vector with attention. We create context vectors for word attention in order to improve our performance.

Word Attention is important because not all words have the same contribution to the representation of the meaning of a sentence. We use the following to extract such words that are important to the meaning of the sentence and aggregate the representation of those to create a sentence vector as follows:

$$u_{it} = tanh(W_w h_{it} + bw) \tag{1}$$

$$\alpha_{it} = \frac{exp(u_{it}^\top u_w)}{\sum_t exp(u_{it}^\top u_w)} \tag{2}$$

$$s_i = \sum_t \alpha_{it} h_{it} \tag{3}$$

Here, u_{it} is a hidden representation of u_{it}. We quantify the importance of the word as the similarity this hidden representation u_{it} with the word level context vector u_w. The normalized importance is α_{it} and this is inputted into a softmax function. Lastly, we compute s_i, the sentence vector as a weighted sum of word annotations based on weights. u_w, the word context vector is also randomly initialized and jointly learned during training.

3.2 MLP with Multi-Channel CNN

The second neural architecture attempted is a MLP with CNN. We begin by defining two channels for our MLP. Channel one consists of a feature dropout layer, a feature map layer, and a flatten layer while channel two consists of just a feature map layer. We don't apply a flatten layer to the second channel as we don't want to reduce the dimensionality. We then concatenate the output of the two channels and apply batch normalization and dropout. The output of these MLP channels is then fed into a CNN architecture as described in the following paper by Kim et al. [16].

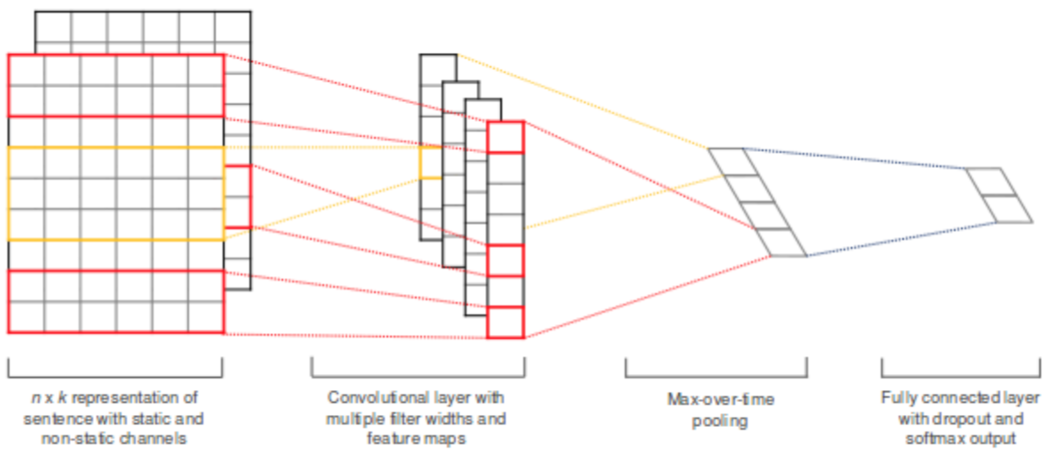

Figure 3: Multi-Channel CNN Architecture.

3.3 AWD-LSTM

The third neural architecture employed was an AWD-LSTM based on the following paper by Howard and Ruder [17]. AWD-LSTM stands for ASGD Weight-Dropped LSTM and provides some substantial benefits to regular Stochastic Gradient Descent (SGD). To better understand this Weight-Dropped LSTM, lets refer to the mathematical formulation of the LSTM,

$$i_t = \sigma(W^i x_t + U^i h_{t-1}) \tag{4}$$

$$f_t = \sigma(W^f x_t + U^f h_{t-1}) \tag{5}$$

$$o_t = \sigma(W^o x_t + U^o h_{t-1}) \tag{6}$$

$$\tilde{c}_t = tanh(W^c x_t + U^c h_{t-1}) \tag{7}$$

$$c_t = i_t \odot \tilde{c} + f_t \odot + \tilde{c}_t - 1 \tag{8}$$

$$h_t = o_t \odot tanh(c_t) \tag{9}$$

where all the W's and U's are weight matrices, x_t is the vector input to time t, h_t is the current exposed hidden state, c_t is the memory cell state and $\odot$ is element-wise multiplication.

In order to reduce overfitting, commonly techniques have acted on the hidden state vector, h_{t-1}, often introducing dropout between timesteps or performing dropout update on the memory state

c_t. We propose the use of DropConnect [21] on the hidden to hidden weight matrices which do not require any modifications to a RNN's formulation. By performing DropConnect on these hidden to hidden weight matrices [U^i, U^f, U^o, U^c] inside the LSTM, we can prevent overfitting on the recurrent connections of the LSTM. This technique could also be used on the non-recurrent weights [W^i, W^f, W^o]. The architecture differences of DropConnect can be seen below:

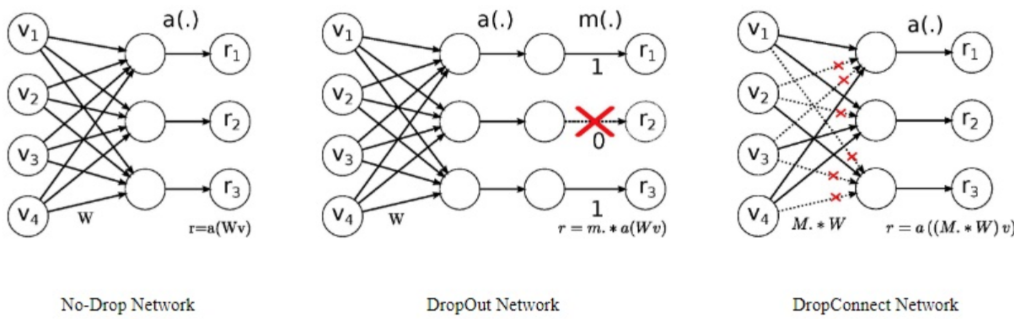

Figure 4: No-Drop Network, DropOut Network, and DropConnect Network architectures respectively.

We use a variant of the ASGD algorithm (Non-monotonically Triggered ASGD) which obviates the need for tuning the averaging trigger T as our optimizer. The algorithm uses a constant learning rate as well throughout which means no tuning is needed. This can be seen below:

Algorithm 1 Non-monotonically Triggered ASGD (NT-ASGD)

Inputs: Initial point w_0, learning rate γ, logging interval L, non-monotone interval n.

```
Initialize k ← 0, t ← 0, T ← 0, logs ← []
while stopping criterion not met do
    Compute stochastic gradient ∇̂f(w_k) and take SGD step (1).
    if mod(k, L) = 0 and T = 0 then
        Compute validation perplexity v.
        if t > n and v > min_{l∈{t−n,···,t}} logs[l] then
            Set T ← k
        end if
        Append v to logs
        t ← t + 1
    end if
end while
```

return $\frac{\sum_{i=T}^{k} w_i}{(k-T+1)}$

Figure 5: NT-ASGD Algorithm

While the algorithm introduces two additional hyperparameters, L for the logging interval and n for the non-monotone interval, research [22] suggests setting L equal to the number of epochs and $n = 5$. We follow a similar preprocessing step as we did for other models above.

3.4 Ensembling

We initially began with even weights for our ensemble, but were able to fine tune this based on validation scores as well as public leaderboard scores. The exact weights are discussed later in the paper.

4 Experiments

In this section we will detail the our feature engineering process, evaluation method, and discuss results.

4.1 Dataset and Feature Engineering

The data we will be using is the Gendered Ambiguous Pronouns (GAP) dataset. It has a test set of 4,000 pairs for test, 4,000 pairs for development, and 908 pairs for validation. The types of features we build are based off the work of Clark & Manning [17] and is as follows:

Embedding Features: Word embeddings of the head word, dependency parent, first word, last word, two preceding words, and the two following words of the mention [17]. The embeddings are then averaged for the five preceding words, five following words, and all the words in the mention, all words in the mention's sentence, and all words in the mention's document.

Additional Mention Features: This refers to the type of mention (pronoun, nominal, proper, or list), the mention's position whether or not the mentions is contained in another mention, and the length of the mention in words [17].

Distance Features: The distance we create is the distance between mentions in sentences, the distance between the mentions in intervening mentions, and whether or not the mentions overlap [17].

Once these features are generated, we concatenate them and create our final dataset. This is then split into train, validation, and test sets so we can feed as input into our neural network.

4.2 Evaluation Method

The evaluation metric for the following project is the multi-class logarithmic loss as shown below.

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} log(p_{ij}) \tag{10}$$

Here, N is the number of samples in the test set, M is 3, log is the natural logarithm, y_{ij} is 1 if observation i belongs to class j and 0 otherwise, p_{ij} is the predicted probability that observation i belongs to class j. The submitted probabilities are not required to sum to one because they are rescaled prior to being scored (each row is divided by the row sum). In the case of extremes of the log function, predicted probabilities are replaced with $max(min(p, 1 - 10^{-15})$.

4.3 Experimental Details

In this section we detail the specific manner in which each experiment was run. We will also provide specific details such as hyperparameters, model configurations, and more. These can be seen in the table below:

Experimental Details			
Model Type	Attention RNN	MLP Multi-Channel CNN	AWD LSTM
Optimizer	Adam	Adam	Adam
Learning Rate	0.001	0.001	0.0003
Batch Size	30	20	64
Epochs	15	10	6

The Attention RNN and MLP Multi-Channel CNN were built on Keras while AWD LSTM was implemented in pytorch. The reasoning behind this was due to prior familiarity with Keras for the first two models and because the AWD LSTM paper used pytorch therefore maintaining consistency.

All the models were trained on our local machine (Macbook Pro, 16 GB RAM). The models all took approximately 30 - 45 minutes to train which is reasonable and reproducible without much effort.

4.4 Results

The first model built was attempting to reproduce the GAP results by Webster et al. [5]. We treat this as our baseline and that scored a logloss of 0.8394. Our Attention RNN improves this score significantly and has currently reached a logloss of 0.7014. Our multi-channel MLP CNN achieves a slightly worse score of 0.7207, but is still a significant improvement over the baseline. Our AWD LSTM is able to achieve a logloss of 0.6897 and performs the best of the three single models.

Our weighted ensemble was able to achieve a score of 0.6201 logloss which for a non BERT method [18] is very competitive. We went with the weights of 0.3 for our Attention RNN, 0.3 for MLP Multi-Channel CNN, and 0.4 for our AWD LSTM. These weights were decided by some manual tuning and testing. Initially even weights were tried, but we noticed improvements by prioritizing AWD LSTM as it had a higher score than the other two models.

5 Analysis

One of the key benefits of ensembling various model types was the ability to be able to generalize to various different types of problems. This means, for example, that despite our Attention RNN being our worst performing single model, it could still capture some behavior which perhaps our AWD-LSTM could not capture despite being the best single model.

We also notice a significant improvement over our initial GAP baseline improving the logloss score by around 0.20. One of the goals of this project was to do this in a reproducible and non computationally intensive manner. Due to the size of our dataset, we were able to achieve this.

6 Conclusion and Future Work

While we were constrained by computational resources and time, we were able to build a competitive ensemble of neural architectures. Our future work would involve some further hyperparameter tuning as well as exploring other popular methods of coreference resolution.

Gender bias is an old problem in the NLP space and this project aimed to tackle the tast of coreference resolution with an eye towards that bias. We hope that the results of this project and improvements in methods to solving this problem will urge other researchers to keep gender bias in mind and account for it while building their datasets and models.

7 Additional Information

My project mentor for this project is Xiaoxue Zang.

References

[1] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 Shared Task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of CoNLL: Shared Task*, pages 1–40, Jeju, Republic of Korea.

[2] Liane Guillou. 2012. Improving pronoun translation for statistical machine translation. In *Proceedings of the Student Research Workshop at the 13th Conference of the EACL*, pages 1–10, Avignon, France.

[3] Kotaro Nakayama. 2008. Wikipedia mining for triple extraction enhanced by co-reference resolution. In *Proceedings of the ISWC Workshop on Social Data on the Web*, Berlin, Germany.

[4] Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of EMNLP*, pages 1971–1982, Seattle, Washington.

[5] Webster, Kellie, Recasens, Marta, Axelrod, Vera, & Baldridge, Jason. (2018) Mind the GAP: A Balanced Corpus of Gendered Ambiguous Pronouns. In *Transactions of the ACL*, page to appear.

[6] Moritz Hardt. 2014. How big data is unfair: Understanding unintended sources of unfairness in data driven decision making. https: //medium.com/@mrtz/how-bigdata-is-unfair-9aa544d739de.

[7] Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd Schema Challenge. In *Proceedings of KR*, pages 552–561, Rome, Italy.

[8] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of NAACL*, New Orleans, Louisiana.

[9] Abbas Ghaddar and Philippe Langlais. 2016a. Coreference in Wikipedia: Main Concept Resolution. In *Proceedings of CoNLL*, pages 229–238, Berlin, Germany.

[10] Altaf Rahman and Vincent Ng. 2012. Resolving complex cases of definite pronouns: The Winograd Schema Challenge. In *Proceedings of EMNLP-CoNLL*, pages 777–789, Jeju, Republic of Korea.

[11] Haoruo Peng, Daniel Khashabi, and Dan Roth. 2015. Solving hard coreference problems. In *Proceedings of NAACL*, pages 809–819, Denver, Colorado.

[12] Trieu H. Trinh and Quoc V. Le. 2018. A simple method for commonsense reasoning. In *ArXiv e-prints v1* https:/arxiv.org/abs/1806.02847.

[13] Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of NAACL*, New Orleans, Louisiana.

[14] Kenton Lee, Luheng He, Mike Lewis, and Luke S. Zettlemoyer. 2017. End-to-end neural coreference resolution. In *EMNLP*.

[15] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

[16] Kim Yoon. (2014) Convolutional Neural Networks for Sentence Classification. In *arXiv preprint arXiv:1408.5882* .

[17] Howard, Jeremy & Ruder, Sebastion. (2018) Universal Language Model Fine-tuning for Text Classification. In *Association for Computational Linguistics (ACL)*.

[18] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *ArXiv e-prints*.