Gendered Ambiguous Pronouns Shared Task: Boosting Model Confidence by Evidence Pooling

Sandeep Attree
New York, NY
sandeep.attree@gmail.com

Abstract

This paper presents a strong set of results for resolving gendered ambiguous pronouns on the Gendered Ambiguous Pronouns shared task. The model presented here draws upon the strengths of state-of-the-art language and coreference resolution models, and introduces a novel evidence-based deep learning architecture. Injecting evidence from the coreference models compliments the base architecture, and analysis shows that the model is not hindered by their weaknesses, specifically gender bias. The modularity and simplicity of the architecture make it very easy to extend for further improvement and applicable to other NLP problems. Evaluation on GAP test data results in a state-of-the-art performance at 92.5% F1 (gender bias of 0.97), edging closer to the human performance of 96.6%. The end-to-end solution[1] presented here placed 1st in the Kaggle competition, winning by a significant lead.

1 Introduction

The Gendered Ambiguous Pronouns (GAP) shared task aims to mitigate bias observed in the performance of coreference resolution systems when dealing with gendered pronouns. State-of-the-art coreference models suffer from a systematic bias in resolving masculine entities more confidently compared to feminine entities. To this end, Webster et al. (2018) published a new GAP dataset[2] to encourage research into building models and systems that are robust to gender bias.

The arrival of modern language models like ELMo (Peters et al., 2018), BERT (Devlin et al., 2018), and GPT (Radford et al., 2018), have significantly advanced the state-of-the art in a wide range of NLP problems. All of them have a common theme in that a generative language model is pretrained on a large amount of data, and is subsequently fine-tuned on the target task data. This approach of transfer learning has been very successful. The current work applies the same philosophy and uses BERT as the base model to encode low-level features, followed by a task-specific module that is trained from scratch (fine-tuning BERT in the process).

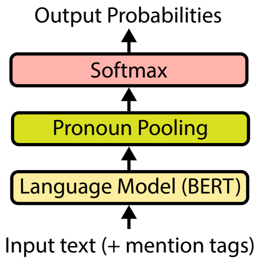


Figure 1: ProBERT: Pronoun BERT. Token embeddings corresponding to the labeled pronoun in the input text are extracted from the last layer of the language model (BERT) and used for prediction.

GAP shared task presents the general GAP problem in *gold-two-mention* (Webster et al., 2018) format and formulates it as a classification problem, where the model must resolve a given pronoun to either of the two given candidates or neither[3]. *Neither* instances are particularly difficult to resolve since they require understanding a wider context and perhaps a knowledge of the world. A parallel for this case can be drawn from Question-Answering systems where identifying unanswerable questions confidently remains an active research area. Recent work shows that it is possible to determine lack of evidence with

[1] The code is available at https://github.com/sattree/gap

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

[3] There is a case in the GAP problem where the pronoun in question may not be coreferent with either of the two mentioned candidates. Such instances will be referred to as *Neither*.

Proceedings of the 1st Workshop on Gender Bias in Natural Language Processing, pages 134–146
Florence, Italy, August 2, 2019.

	M	F	Total
gap-development	1000	1000	2000
gap-validation	227	227	454
gap-test	1000	1000	2000
gpr-neither (section 2.1)	129	124	253
stage 2 test (kaggle)†	6499	5860	12359

Table 1: Corpus statistics. Masculine (M) and Feminine (F) instances were identified based on the gender of the pronoun mention labeled in the sample. †Only a subset of these may have been used for final evaluation.

greater confidence by explicitly modeling for it. Works of Zhong et al. (2019) and Kundu and Ng (2018) demonstrate model designs with specialized deep learning architectures that encode evidence in the input and show significant improvement in identifying unanswerable questions. This paper first introduces a baseline that is based on a language model. Then, a novel architecture for pooling evidence from off-the-shelf coreference models is presented, that further boosts the confidence of the base classifier and specifically helps in resolving *Neither* instances. The main contributions of this paper are:

- Demonstrate the effectiveness of pretrained language models and their transferability to establish a strong baseline (ProBERT) for the *gold-two-mention* shared task.
- Introduce an Evidence Pooling based neural architecture (GREP) to draw upon the strengths of off-the-shelf coreference systems.
- Present the model results that placed 1st in the GAP shared task Kaggle competition.

2 Data and Preprocessing

Table 1 shows the data distribution. All datasets are approximately gender balanced, other than stage 2 test set. The datasets, gap-development, gap-validation, and gap-test, are part of the publicly available GAP corpus. The preprocessing and sanitization steps are described next.

2.1 Data Augmentation: *Neither* instances

In an attempt to upsample and boost the classifier's confidence in the underrepresented $Neither$ category (Table 2), about 250 instances were added manually. These were created by obtaining cluster predictions from the coreference model by Lee et al. (2018) and choosing a pronoun and the two candidate entities A and B from disjoint clusters. However, in the interest of time, this strategy was not fully pursued. Instead, the evidence pooling module was used to resolve this problem, as will become clear from the discussion in section 6.

2.2 Mention Tags

The raw text snippet is manipulated by enclosing the labeled span of mentions with their associated tags, i.e. <P> for pronoun, <A> for entity mention A, and <B> for entity mention B. The primary reason for doing this is to provide the positional information of the labeled mentions implicitly within the text as opposed to explicitly through additional features. A secondary motivation was to test the language model's sensitivity to noise in input text structure, and its ability to adapt the pronoun representation to the positional tags. Figure 2 shows an example of this annotation scheme.

> ... NHLer Gary Suter and Olympic-medalist <**A**> Bob Suter <**A**> are <**B**> Dehner <**B**>'s uncles. <**P**> His <**P**> cousin is Minnesota Wild's alternate captain Ryan ...

Figure 2: Sample text-snippet after annotating the mentions with their corresponding tags. Bob Suter and Dehner were tagged as entities A and B, and the mention 'His' following them was tagged as the pronoun.

2.3 Label Sanitization

Only samples where labels can be corrected unambiguously based on snippet-context were corrected[4]. The Wikipedia page-context and url-context were not used. A visualization tool [5] was also developed as part of this work to aid in this activity. Table 2 lists the corpus statistics before and after the sanitization process.

2.4 Coreference Signal

Transformer networks have been found to have limited capability in modeling long-range dependency (Dai et al., 2018; Khandelwal et al., 2018). It has also been noticed in the past that the coreference problem benefits significantly from global

[4] Corrected labels can be found at `https://github.com/sattree/gap`. This set was generated independently to avoid any unintended bias. More sets of corrections can be found at `https://www.kaggle.com/c/gendered-pronoun-resolution/discussion/81331#503094`

[5] `https://github.com/sattree/gap/visualization`

	Before sanitization			After sanitization			
	A	B	NEITHER	A	B	NEITHER	Total
gap-development	874	925	201	857(-37)(+20)	919(-32)(+26)	224(-4)(+27)	2000
gap-validation	187	205	62	184(-10)(+7)	206(-7)(+8)	64(-4)(+6)	454
gap-test	918	855	227	894(-42)(+18)	860(-27)(32)	246(-8)(27)	2000

Table 2: GAP dataset label distribution before and after sanitization. (-x) indicates the number of samples that were moved out of a given class and (+x) indicates the number of samples that were added post-sanitization.

knowledge (Lee et al., 2018). Being cognizant of these two factors, it would be useful to inject predictions from off-the-shelf coreference models as an auxiliary source of evidence (with input text context being the primary evidence source). The models chosen for this purpose are Parallelism+URL (Webster et al., 2018), AllenNLP[6], NeuralCoref[7], and e2e-coref (Lee et al., 2018).

3 Model Architecture

3.1 ProBERT: baseline model

ProBERT uses a fine-tuned BERT language model (Devlin et al., 2018; Howard and Ruder, 2018) with a classification head on top to serve as baseline. The snippet-text is augmented with mention-level tags (section 2.2) to capture the positional information of the pronoun, entity A, and entity B mentions, before feeding the text as input to the model. Position-wise token representation corresponding to the pronoun is extracted from the last layer of the language model. With GAP dataset and WordPiece tokenization (Devlin et al., 2018), all pronouns were found to be single token entities.

Let $E_p \in \mathbb{R}^H$ (where H is the dimensionality of the language model output) denote the pooled pronoun vector. A linear transformation is applied to it, followed by softmax, to obtain a probability distribution over classes A, B, and NEITHER, $P = softmax(W^T E_p)$, where $W \in \mathbb{R}^{H \times 3}$ is the linear projection weight matrix. All the parameters are jointly trained to minimize cross entropy loss. This simple architecture is depicted in Figure 1. Only $H \times 3$ new parameters are introduced in the architecture, allowing the model to use training data more efficiently.

A natural question arises as to why this model functions so well (see section 5.2) with just the pronoun representation. This is discussed in section 6.1.

3.2 GREP: Gendered Resolution by Evidence Pooling

The architecture for GREP pairs the simple ProBERT architecture with a novel Evidence Pooling module. The Evidence Pooling (EP) module leverages cluster predictions from pretrained (or heuristics-based) coreference models to gather evidence for the resolution task. The internals of the coreference models are opaque to the system, allowing for any evidence source such as a knowledge base to be included as well. This design choice limits us from propagating the gradients through the coreference models, thereby losing information and leaving them noisy. The difficulty of efficiently training deeper architectures paired with the noisy cluster predictions (the best coreference model has an $F1$ performance of only 64% on gap-test) makes this a challenging design problem. The EP module uses self-attention mechanism described in Vaswani et al. (2017) to compute the compatibility of cluster mentions with respect to the pronoun and the two candidates, entity A, and entity B. The simple and easily extensible architecture of this module is described next.

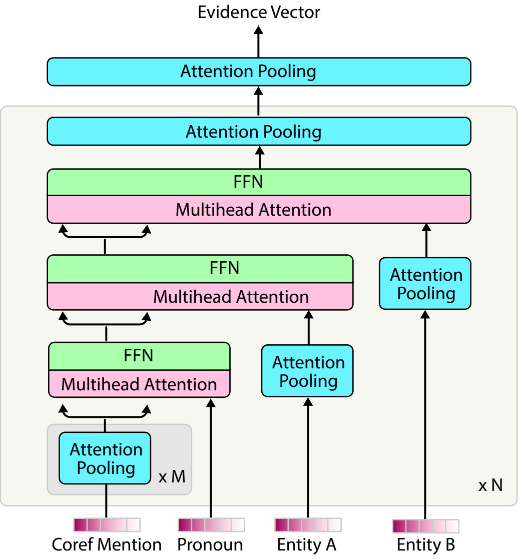


Figure 3: Evidence Pooling module architecture

[6] https://allennlp.org/models

[7] https://github.com/huggingface/neuralcoref

Suppose we have access to N off-the-shelf coreference models and each predicts T_n mentions that are coreferent with the given pronoun. Let P, A, and B, refer to the mentioned entities labeled in the text-snippet as the pronoun and entities A and B, respectively. Without loss of generality, let us consider the *nth* coreference model and *mth* mention in the cluster predicted by it. Let $E_m \in \mathbb{R}^{T_m \times H}$, $E_p \in \mathbb{R}^{T_p \times H}$, $E_a \in \mathbb{R}^{T_a \times H}$ and $E_b \in \mathbb{R}^{T_b \times H}$, denote the position-wise token embeddings obtained from the last layer of the language model for each of the mentions, where T is the number of tokens in each mention. The first step is to aggregate the information at the mention-level. Self-attention is used to reduce the mention tokens, an operation that will be referred to as $AttnPool$ (attention pooling) hereafter. A single layer MLP is applied to compute position-wise compatibility score, which is then normalized and used to compute a weighted average over the mention tokens for a pooled representation of the mention as follows:

$$M_m = tanh(E_m W_m + b_m) \in \mathbb{R}^{T_m \times H} \quad (1)$$

$$a_m = softmax(M_m) \in \mathbb{R}^{T_m} \quad (2)$$

$$AttnPool(E_m, W_m) = A_m = \sum_{i}^{T_m} a_m E_m \in \mathbb{R}^H \quad (3)$$

Similarly, a pooled representation of all mentions in the cluster predicted by the nth coreference model, and of P, A, and B entity mentions is obtained. Let $A_n \in \mathbb{R}^{T_n \times H}$ denote the joint representation of cluster mentions, and A_p, A_a, and A_b, the pooled representations of entity mentions. Next, to compute the compatibility of the cluster with respect to the given entities, we systematically transform the cluster representation by passing it through a transformer layer (Vaswani et al., 2017). A sequence of such transformations is applied successively by feeding A_p, A_a, and A_b as query vectors at each stage. Each such transformer layer consists of a multi-head attention and feed-forward (FFN) projection layers. The reader is referred to Vaswani et al. (2017) for further information on $MultiHead$ operation.

$$FFN(x) = tanh(W_x x + b_x) \in \mathbb{R}^{T_p \times H} \quad (4)$$

$$C_p = FFN(MultiHead(A_p, A_m, A_m)) \in \mathbb{R}^{T_p \times H} \quad (5)$$

$$C_a = FFN(MultiHead(A_a, C_p, C_p)) \in \mathbb{R}^{T_a \times H} \quad (6)$$

$$C_b = FFN(MultiHead(A_b, C_a, C_a)) \in \mathbb{R}^{T_b \times H} \quad (7)$$

The transformed cluster representation C_b is then reduced at the cluster-level and finally at the coreference model level by attention pooling as:

$$A_c = AttnPool(C_b, W_c) \in \mathbb{R}^{N \times H} \quad (8)$$

$$A_{co} = AttnPool(A_c, W_{co}) \in \mathbb{R}^H \quad (9)$$

A_{co} represents the evidence vector that encodes information obtained from all the coreference models. Finally, the evidence vector is concatenated with the pronoun representation, and is once again fed through a linear layer and softmax to obtain class probabilities.

$$C = [E_p; A_{co}] \in \mathbb{R}^{2H} \quad (10)$$

$$P = softmax(W^T C + b) \in \mathbb{R}^3 \quad (11)$$

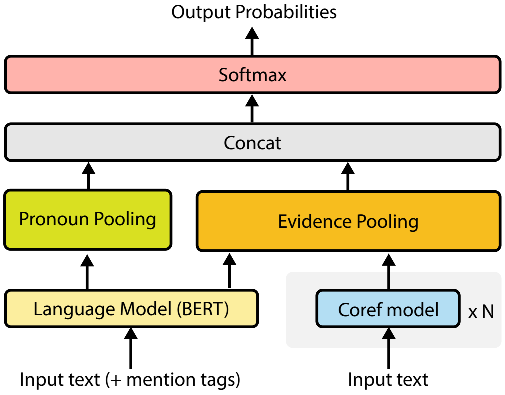


Figure 4: GREP model architecture

The end-to-end GREP model architecture is illustrated in Figure 4.

4 Training

All models were trained on 4 NVIDIA V100 GPUs (16GB memory each). The *pytorch-pretrained-bert*[8] library was used as the language model module and saved model checkpoints were used for initialization. Adam (Kingma and Ba, 2014) optimizer was used with $\beta_1 = 0.9$, $\beta_2 =$

[8] https://github.com/huggingface/pytorch-pretrained-BERT/. BertTokenizer from this package was used for tokenization of the text. BertAdam was used as the optimizer. This package contains resources for all variants of BERT, i.e. bert-base-uncased, bert-base-cased, bert-large-uncased and bert-large-cased.

	F1				logloss
	M	F	B	O	
Lee et al. (2017)†	67.7	60.0	0.89	64.0	-
Parallelism†	69.4	64.4	0.93	66.9	-
Parallelism+URL†	72.3	68.8	0.95	70.6	-
RefReader, LM & coref‡	72.8	71.4	**0.98**	72.1	-
ProBERT (bert-base-uncased)	88.9	86.7	**0.98**	87.8	.382
GREP (bert-base-uncased)	90.4	87.6	0.97	89.0	.350
ProBERT (bert-large-uncased)	90.8	88.6	**0.98**	89.7	.376
GREP (bert-large-uncased)	**94.0**	**91.1**	0.97	**92.5**	**.317**
Human Performance (estimated)	97.2	96.1	0.99	96.6	-

Table 3: Single model performance on gap-test set by gender. M: masculine, F: feminine, B: (bias) ratio of feminine to masculine performance, O: overall. Log loss is not available for systems that only produce labels. †As reported by Webster et al. (2018). ‡As reported by Liu et al. (2019), their model does not use *gold-two-mention* labeled span information for prediction.

0.999, $\epsilon = 1e^{-6}$, and a fixed learning rate of $4e^{-6}$. For regularization, a fixed dropout (Srivastava et al., 2014) rate of 0.1 was used in all layers and a weight decay of 0.01 was applied to all parameters. Batch sizes of 16 and 8 samples were used for model variants with bert-base and bert-large respectively. Models with bert-base took about 6 mins to train while those with bert-large took up to 20 mins.

For single model performance evaluation, the models were trained on gap-train, early-stopping was based off of gap-validation, and gap-test was used for test evaluation. Kaggle competition results were obtained by training models on all datasets, i.e. gap-train, gap-validation, gap-test, and gpr-neither (a total of 4707 samples), in a 5-Fold Cross-Validation (Friedman et al., 2001) fashion. Each model gets effectively trained on 3768 samples, while 942 samples were held-out for validation. Training would terminate upon identifying an optimal early stopping point based on performance on the validation set with an evaluation frequency of 80 gradient steps. Model's access is limited to snippet-context, and the Wikipedia page-context is not used. However, page-url context may be used via coreference signal (Parallelism+URL).

5 Results

The performance of ProBERT and GREP models is benchmarked against results previously established by Webster et al. and Liu et al. (2019). It is worth noting that Liu et al. do not use *gold-two-mention* labeled spans for prediction and hence their results may not be directly comparable. This section first introduces an estimate of human performance on this task. Then, results for single model performance are presented, followed by ensembled model results that won the Kaggle competition. $F1$ performance scores were obtained by using the GAP scorer script[9] provided by Webster et al.. Wherever applicable, log loss (the official Kaggle metric) performance is reported as well.

5.1 Human Performance

Errors found in crowd-sourced labels are considered a measure of human performance on this task, and serve as a benchmark. The corrections are only a best-effort attempt to fix some obvious mistakes found in the dataset labels, and were made with certain considerations (section 2.3). This performance measure is subject to variation based on an evaluator's opinion on ambiguous samples.

5.2 Single Model Performance

Single model performance on GAP test set is shown in Table 3. The GREP model (with bert-large-uncased as the language model) achieves a powerful state-of-the-art performance on this task. The model significantly benefits from evidence pooling, gaining 6 points in terms of log loss and 2.8 points in $F1$ accuracy. Further analysis of the source of these gains is discussed in section 6.

While it may seem that the significantly improved performance of GREP has been achieved

[9] https://github.com/google-research-datasets/gap-coreference/blob/master/gap_scorer.py

Model	Dataset	F1				logloss
		M	F	B	O	
LM=bert-large-uncased, seed=42	OOF all	94.3	93.21	0.99	93.8	.261
	OOF gap-test	94.2	93.7	0.99	93.9	.254
LM=bert-large-cased, seed=42	OOF all	94.3	93.9	0.99	94.1	.249
	OOF gap-test	94.3	93.5	0.99	93.9	.242
Ensemble: (LM=bert-large-uncased + seeds=42,59,75,46,91)	OOF all	94.8	94.2	0.99	94.5	.195
	OOF gap-test	94.5	94.33	1.00	94.4	.193
Ensemble: (LM=bert-large-cased + seeds=42,59,75,46,91)	OOF all	95.1	94.4	0.99	94.7	.187
	OOF gap-test	94.9	94.1	0.99	94.4	.183
Ensemble: (LM=bert-large-uncased, bert-large-cased + seeds=42,59,75,46,91)	OOF all	95.3	94.7	0.99	95.0	.176
	OOF gap-test	95.1	94.7	1.00	94.9	.175
	Stage 2 test	-	-	-	-	**.137**†

Table 4: GREP model performance results in the Kaggle competition. Out-of-fold (OOF) error is reported on all data, i.e. gap-development, gap-validation, gap-test, and gpr-neither, as well as on gap-test explicitly for comparison against single model performance results. Since early stopping is based on OOF samples, OOF errors reported here cannot be considered as an estimate of test error. Nevertheless, stage 2 test performance benchmarks the model. †Due to a bug, the model did not fully leverage coref evidence, further gains are expected with the fixed version.

at a small cost in terms of gender bias, an attentive reader would realize that the model enjoys improved performance for both genders. Performance gains in masculine instances are much higher compared to feminine instances, and the slight degradation in bias ratio is a manifestation of this. The superior performance of GREP provides evidence that for a given sample context, the model architecture is able to successfully discriminate between the coreference signals, and identify their usefulness.

5.3 Kaggle Competition[10]

To encourage fairness in modeling, the competition was organized in two stages. This strategy eliminates any attempts at leaderboard probing and other such malpractices. Furthermore, models were frozen at the end of stage 1 and were only allowed to operate in inference mode to generate predictions for stage 2 test submission. Additionally, no feedback was provided on stage 2 submissions (in terms of performance score) until the end of the competition.

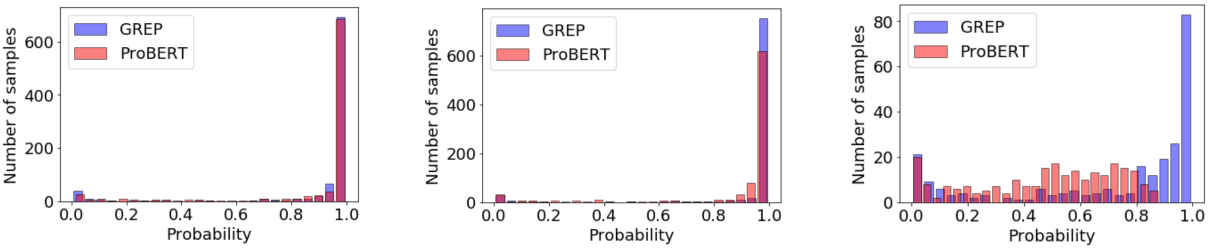
GREP model is trained as described in section 4 and out-of-fold (oof) error on the held-out samples is reported. The experiments are repeated with 5 different random seeds (42, 59, 75, 46, 91) for initialization. Finally, two sets of models are trained with bert-large-uncased and bert-large-cased as the language models. The overall scheme leads to 50 models being trained in total, and 50 sets of predictions being generated on stage 2 test data. To generate predictions for submission, ensembling is done by simply taking the unbiased weighted mean over the 50 individual prediction sets.

Table 4 presents a granular view of the winning model performance. This performance comes very close to human performance and has almost no gender bias. As the table shows, the ensemble models achieve much larger gains in log loss as compared to $F1$ accuracy. This is expected since the committee of models makes more confident decisions on "easier" examples. Two insights can be drawn by comparing these results with the single model performance presented in section 5.2: (1) model accuracy benefits from more training data, although the gains are marginal at best (92.5 vs 93.9) given that the model was trained on approximately twice the amount of data; (2) ensembling has a similar effect as evidence pooling, i.e., models become more confident in their predictions.

6 Discussion

Results shown in section 5 establish the superior performance of GREP compared to ProBERT. This can be attributed to two sources: (1) GREP corrects some errors made by ProBERT, reflected in F1; and (2) where predictions are correct, GREP

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



(a) A

(b) B

(c) NEITHER

Figure 5: Comparison of probabilities assigned by ProBERT and GREP. Figures show distribution of predicted class probabilities assigned by the models to samples from that class.

		GREP	
	ProBERT	Incorrect	Correct
A	Incorrect	44	38
	Correct	28	784
B	Incorrect	37	39
	Correct	9	775
NEITHER	Incorrect	45	44
	Correct	11	146
Overall	Incorrect	126	121
	Correct	48	1705

Table 5: Class-wise comparison of model accuracy for ProBERT and GREP. Off-diagonal terms show cases where GREP fixes errors made by ProBERT and vice-versa.

is more confident in its predictions, reflected in log loss. To investigate this, error analysis is performed on gap-test.

Figure 5 shows a class-wise comparison of probabilities generated by the two models. It can be seen that GREP is more confident in its predictions (all distributions appear translated closer to 1.0), and the improvement is overwhelmingly evident for the NEITHER class. To understand the difference between the two models, confusion matrix statistics are presented in table 5. The diagonal terms show the number of instances that the two models agree on, and the off-diagonal terms show where they disagree. The numbers reveal that the evidence pooling module not only boosts the model confidence but also helps in correctly resolving *Neither* instances (44 vs 11), indicating that the model is successfully able to build evidence for or against the given candidates.

Appendix A details the behavior of GREP through some examples. The first example is particularly interesting - while it is trivial for a human to resolve this, a machine would require knowledge of the world to understand "death" and its implications.

6.1 Unreasonable Effectiveness of ProBERT

It would seem unreasonable that ProBERT is able to perform so well with the noisy input text (due to mention tags) and is able to make the classification decision by looking at the pronoun alone. The following two theories may explain this behavior: (1) attention heads in the (BERT) transformer architecture are able to specialize the pronoun representation in the presence of the supervision signal; (2) the special nature of dropout (present in every layer) makes the model immune to a small amount of noise, and at the same time prevents the model from ignoring the tags. The analysis of attention heads to investigate these claims should form the scope of future work.

7 Conclusion

A powerful set of results have been established for the shared task. Work presented in this paper makes it feasible to efficiently employ neural attention for pooling information from auxiliary sources of global knowledge. The evidence pooling mechanism introduced here is able to leverage upon the strengths of off-the-shelf coreference solvers without being hindered by their weaknesses (gender bias). A natural extension of the GREP model would be to solve the gendered pronoun resolution problem beyond the scope of the *gold-two-mention* task, i.e., without accessing the labeled gold spans.

Acknowledgments

I would like to thank Google AI Language and Kaggle for hosting and organizing this competition, and for providing a platform for independent research.

References

Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2018. Transformer-xl: Language modeling with longer-term dependency.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2001. *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. *arXiv preprint arXiv:1805.04623*.

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

Souvik Kundu and Hwee Tou Ng. 2018. A nil-aware answer extraction framework for question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4243–4252.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. *arXiv preprint arXiv:1804.05392*.

Fei Liu, Luke Zettlemoyer, and Jacob Eisenstein. 2019. The referential reader: A recurrent entity network for anaphora resolution. *arXiv preprint arXiv:1902.01541*.

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

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the gap: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.

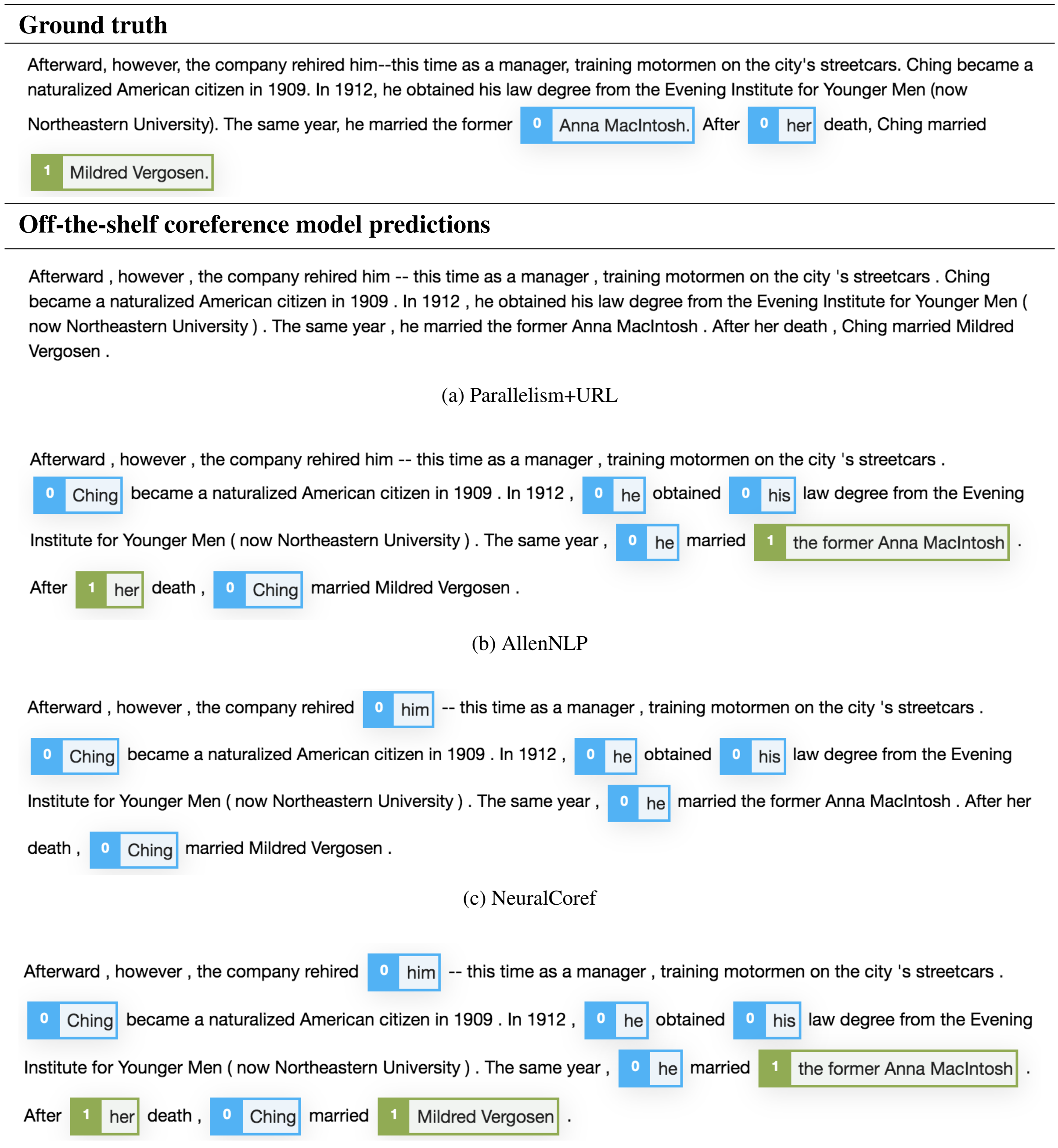
Victor Zhong, Caiming Xiong, Nitish Shirish Keskar, and Richard Socher. 2019. Coarse-grain fine-grain coattention network for multi-evidence question answering. *arXiv preprint arXiv:1901.00603*.

A Examples

Tables 6, 7, 8, and Figure 6 show an example of how incorporating evidence from the coreference models helps GREP to correct a prediction error made by ProBERT. While the example is trivial for a human to resolve, a machine would require knowledge of the world to understand “death” and its implications. ProBERT is unsure about the resolution and ends up assigning comparable probabilities to both entities A and B. GREP, on the other hand, is able to shift nearly all the probability mass from B to the correct resolution A, in light of strong evidence presented by the coreference solvers. Figure 6 illustrates an interesting phenomenon; while e2e-coref groups the pronoun and both entities A and B in the same cluster, the model architecture is able to harvest information from AllenNLP predictions, propagating the belief that entity A must be the better candidate. The above observations indicate that by pooling evidence from various sources, the model is able to reason over a larger space and build a rudimentary form of world knowledge.

Tables 9, 10, 11, and Figure 7 show a second example. This example is not easy even for a human to resolve without reading and understanding the full context. A model may find this situation to be adverse given the presence of too many named entities as distractor elements; and the url-context can be misleading since the pronoun referent is not the subject of the article. Nevertheless, the model is able to successfully build evidence against the given candidates, and resolve with a very high confidence of 92.5%.

Finally, a third example is shown in Tables 12 and 13. This example shows that the model doesn’t simply make a majority decision, rather considers interactions between the global structure exposed by the various evidence sources.



Ground truth

Afterward, however, the company rehired him--this time as a manager, training motormen on the city's streetcars. Ching became a naturalized American citizen in 1909. In 1912, he obtained his law degree from the Evening Institute for Younger Men (now Northeastern University). The same year, he married the former 0 Anna MacIntosh. After 0 her death, Ching married 1 Mildred Vergosen.

Off-the-shelf coreference model predictions

Afterward , however , the company rehired him -- this time as a manager , training motormen on the city 's streetcars . Ching became a naturalized American citizen in 1909 . In 1912 , he obtained his law degree from the Evening Institute for Younger Men (now Northeastern University) . The same year , he married the former Anna MacIntosh . After her death , Ching married Mildred Vergosen .

(a) Parallelism+URL

Afterward , however , the company rehired him -- this time as a manager , training motormen on the city 's streetcars . 0 Ching became a naturalized American citizen in 1909 . In 1912 , 0 he obtained 0 his law degree from the Evening Institute for Younger Men (now Northeastern University) . The same year , 0 he married 1 the former Anna MacIntosh . After 1 her death , 0 Ching married Mildred Vergosen .

(b) AllenNLP

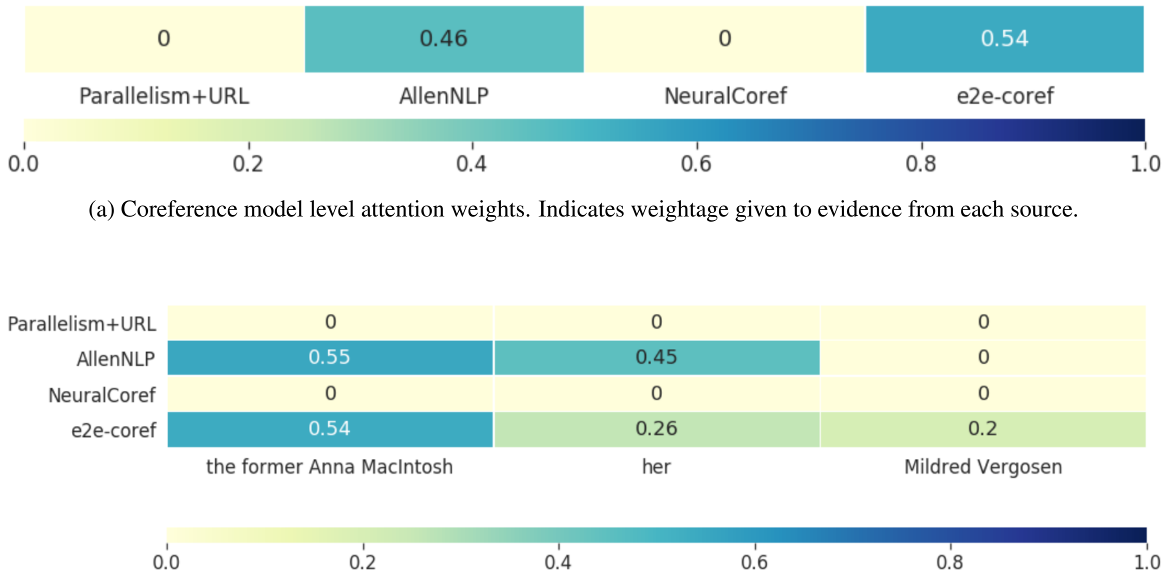
Afterward , however , the company rehired 0 him -- this time as a manager , training motormen on the city 's streetcars . 0 Ching became a naturalized American citizen in 1909 . In 1912 , 0 he obtained 0 his law degree from the Evening Institute for Younger Men (now Northeastern University) . The same year , 0 he married the former Anna MacIntosh . After her death , 0 Ching married Mildred Vergosen .

(c) NeuralCoref

Afterward , however , the company rehired 0 him -- this time as a manager , training motormen on the city 's streetcars . 0 Ching became a naturalized American citizen in 1909 . In 1912 , 0 he obtained 0 his law degree from the Evening Institute for Younger Men (now Northeastern University) . The same year , 0 he married 1 the former Anna MacIntosh . After 1 her death , 0 Ching married 1 Mildred Vergosen .

(d) e2e-coref (Lee et al., 2018)

Table 6: Example 1 - Illustration of ground truth and coreference model predictions. Mentions belonging to a coreference cluster are color coded and indexed. Visualizations were produced using the code module at https://github.com/sattree/gap/visualization.



(a) Coreference model level attention weights. Indicates weightage given to evidence from each source.

(b) Cluster mention level attention weights. Indicates weightage given to each mention within an evidence cluster.

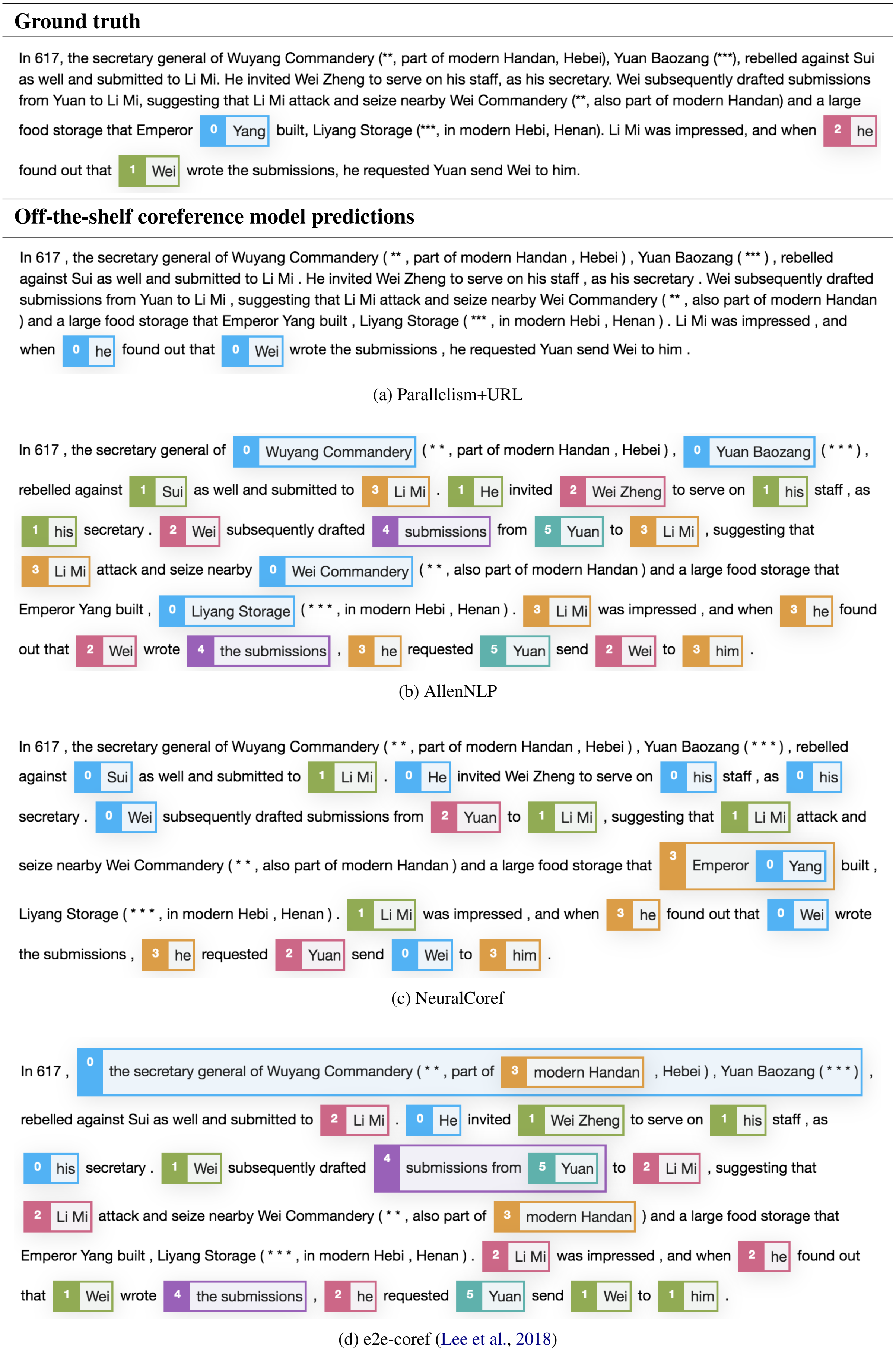
Figure 6: Example 1 - Visualization of normalized attention scores assigned by the hierarchical attention pooling layers in the evidence pooling module

id	Pronoun	Pronoun offset	A	A offset	A coref	B	B offset	B coref	Url
test-282	her	410	Anna MacIntosh	338	True	Mildred Vergosen	475	False	http://en.wikipedia.org/wiki/Cyrus_S._Ching

Table 7: Example 1 - Sample details from GAP test set.

	P(A)	P(B)	P(NEITHER)
ProBERT	0.405	0.452	0.142
GREP	0.718	0.038	0.244

Table 8: Example 1 - A comparison of probabilities assigned by ProBERT and GREP



Ground truth

In 617, the secretary general of Wuyang Commandery (**, part of modern Handan, Hebei), Yuan Baozang (***), rebelled against Sui as well and submitted to Li Mi. He invited Wei Zheng to serve on his staff, as his secretary. Wei subsequently drafted submissions from Yuan to Li Mi, suggesting that Li Mi attack and seize nearby Wei Commandery (**, also part of modern Handan) and a large food storage that Emperor 0 Yang built, Liyang Storage (***, in modern Hebi, Henan). Li Mi was impressed, and when 2 he found out that 1 Wei wrote the submissions, he requested Yuan send Wei to him.

Off-the-shelf coreference model predictions

In 617 , the secretary general of Wuyang Commandery (** , part of modern Handan , Hebei) , Yuan Baozang (***) , rebelled against Sui as well and submitted to Li Mi . He invited Wei Zheng to serve on his staff , as his secretary . Wei subsequently drafted submissions from Yuan to Li Mi , suggesting that Li Mi attack and seize nearby Wei Commandery (** , also part of modern Handan) and a large food storage that Emperor Yang built , Liyang Storage (*** , in modern Hebi , Henan) . Li Mi was impressed , and when 0 he found out that 0 Wei wrote the submissions , he requested Yuan send Wei to him .

(a) Parallelism+URL

In 617 , the secretary general of 0 Wuyang Commandery (* * , part of modern Handan , Hebei) , 0 Yuan Baozang (* * *) , rebelled against 1 Sui as well and submitted to 3 Li Mi . 1 He invited 2 Wei Zheng to serve on 1 his staff , as 1 his secretary . 2 Wei subsequently drafted 4 submissions from 5 Yuan to 3 Li Mi , suggesting that 3 Li Mi attack and seize nearby 0 Wei Commandery (* * , also part of modern Handan) and a large food storage that Emperor Yang built , 0 Liyang Storage (* * * , in modern Hebi , Henan) . 3 Li Mi was impressed , and when 3 he found out that 2 Wei wrote 4 the submissions , 3 he requested 5 Yuan send 2 Wei to 3 him .

(b) AllenNLP

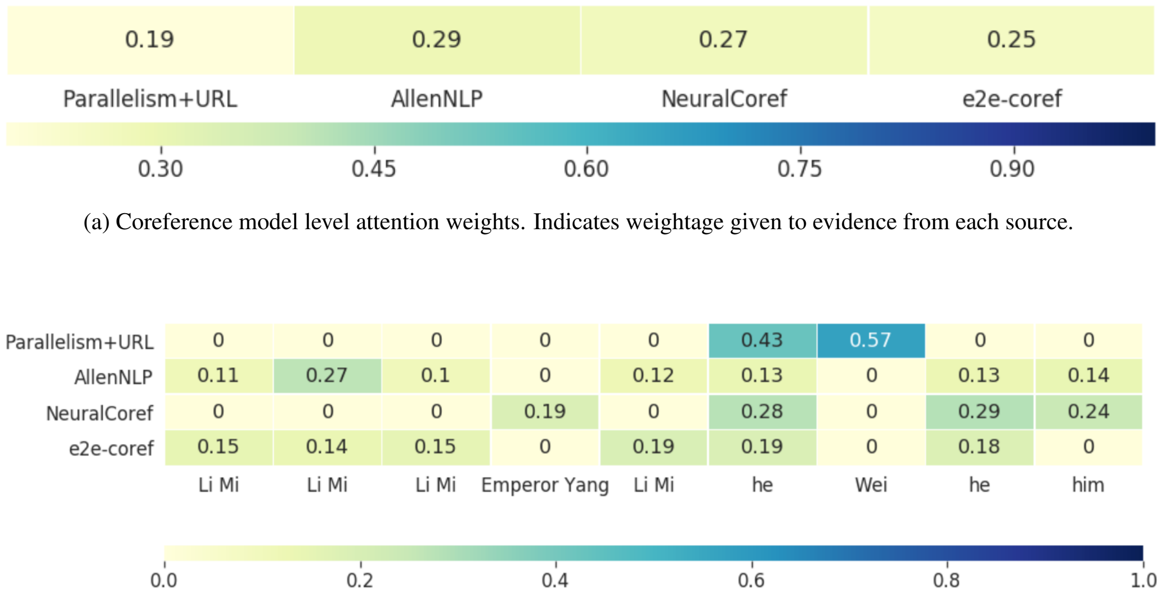
In 617 , the secretary general of Wuyang Commandery (* * , part of modern Handan , Hebei) , Yuan Baozang (* * *) , rebelled against 0 Sui as well and submitted to 1 Li Mi . 0 He invited Wei Zheng to serve on 0 his staff , as 0 his secretary . 0 Wei subsequently drafted submissions from 2 Yuan to 1 Li Mi , suggesting that 1 Li Mi attack and seize nearby Wei Commandery (* * , also part of modern Handan) and a large food storage that 3 Emperor 0 Yang built , Liyang Storage (* * * , in modern Hebi , Henan) . 1 Li Mi was impressed , and when 3 he found out that 0 Wei wrote the submissions , 3 he requested 2 Yuan send 0 Wei to 3 him .

(c) NeuralCoref

In 617 , 0 the secretary general of Wuyang Commandery (* * , part of 3 modern Handan , Hebei) , Yuan Baozang (* * *) , rebelled against Sui as well and submitted to 2 Li Mi . 0 He invited 1 Wei Zheng to serve on 1 his staff , as 0 his secretary . 1 Wei subsequently drafted 4 submissions from 5 Yuan to 2 Li Mi , suggesting that 2 Li Mi attack and seize nearby Wei Commandery (* * , also part of 3 modern Handan) and a large food storage that Emperor Yang built , Liyang Storage (* * * , in modern Hebi , Henan) . 2 Li Mi was impressed , and when 2 he found out that 1 Wei wrote 4 the submissions , 2 he requested 5 Yuan send 1 Wei to 1 him .

(d) e2e-coref (Lee et al., 2018)

Table 9: Example 2 - Illustration of ground truth and coreference model predictions. Mentions belonging to a coreference cluster are color coded and indexed.



(a) Coreference model level attention weights. Indicates weightage given to evidence from each source.

(b) Cluster mention level attention weights. Indicates weightage given to each mention within an evidence cluster.

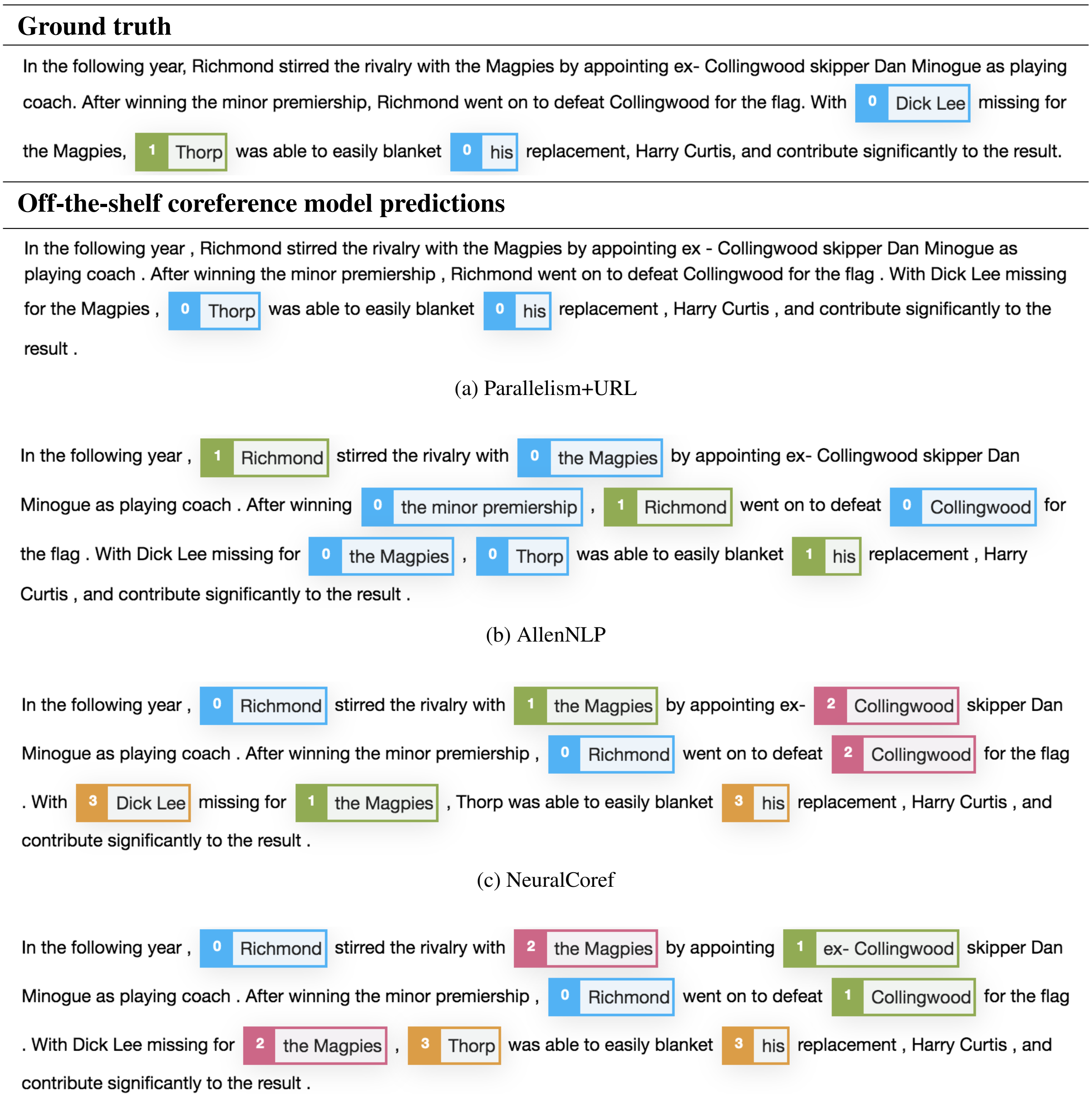
Figure 7: Example 2 - Visualization of normalized attention scores assigned by the hierarchical attention pooling layers in the evidence pooling module

id	Pronoun	Pronoun offset	A	A offset	A coref	B	B offset	B coref	Url
test-406	he	803	Yang	636	False	Wei	916	False	http://en.wikipedia.org/wiki/Wei_Zheng

Table 10: Example 2 - Sample details from GAP test set.

	P(A)	P(B)	P(NEITHER)
ProBERT	0.790	0.038	0.172
GREP	0.055	0.020	0.925

Table 11: Example 2 - A comparison of probabilities assigned by ProBERT and GREP



Ground truth

In the following year, Richmond stirred the rivalry with the Magpies by appointing ex- Collingwood skipper Dan Minogue as playing coach. After winning the minor premiership, Richmond went on to defeat Collingwood for the flag. With 0 Dick Lee missing for the Magpies, 1 Thorp was able to easily blanket 0 his replacement, Harry Curtis, and contribute significantly to the result.

Off-the-shelf coreference model predictions

In the following year , Richmond stirred the rivalry with the Magpies by appointing ex - Collingwood skipper Dan Minogue as playing coach . After winning the minor premiership , Richmond went on to defeat Collingwood for the flag . With Dick Lee missing for the Magpies , 0 Thorp was able to easily blanket 0 his replacement , Harry Curtis , and contribute significantly to the result .

(a) Parallelism+URL

In the following year , 1 Richmond stirred the rivalry with 0 the Magpies by appointing ex- Collingwood skipper Dan Minogue as playing coach . After winning 0 the minor premiership , 1 Richmond went on to defeat 0 Collingwood for the flag . With Dick Lee missing for 0 the Magpies , 0 Thorp was able to easily blanket 1 his replacement , Harry Curtis , and contribute significantly to the result .

(b) AllenNLP

In the following year , 0 Richmond stirred the rivalry with 1 the Magpies by appointing ex- 2 Collingwood skipper Dan Minogue as playing coach . After winning the minor premiership , 0 Richmond went on to defeat 2 Collingwood for the flag . With 3 Dick Lee missing for 1 the Magpies , Thorp was able to easily blanket 3 his replacement , Harry Curtis , and contribute significantly to the result .

(c) NeuralCoref

In the following year , 0 Richmond stirred the rivalry with 2 the Magpies by appointing 1 ex- Collingwood skipper Dan Minogue as playing coach . After winning the minor premiership , 0 Richmond went on to defeat 1 Collingwood for the flag . With Dick Lee missing for 2 the Magpies , 3 Thorp was able to easily blanket 3 his replacement , Harry Curtis , and contribute significantly to the result .

(d) e2e-coref (Lee et al., 2018)

Table 12: Example 3 - Illustration of ground truth and coreference model predictions. Mentions belonging to a coreference cluster are color coded and indexed.

	P(A)	P(B)	P(NEITHER)
ProBERT	0.028	0.968	0.003
GREP	0.724	0.263	0.012

Table 13: Example 3 - A comparison of probabilities assigned by ProBERT and GREP