Incorporating Syntax and Semantics in Coreference Resolution with Heterogeneous Graph Attention Network

Fan Jiang and **Trevor Cohn**
School of Computing and Information Systems
The University of Melbourne, Victoria, Australia
fan.jiang1@student.unimelb.edu.au
t.cohn@unimelb.edu.au

Abstract

External syntactic and semantic information has been largely ignored by existing neural coreference resolution models. In this paper, we present a heterogeneous graph-based model to incorporate syntactic and semantic structures of sentences. The proposed graph contains a syntactic sub-graph where tokens are connected based on a dependency tree, and a semantic sub-graph that contains arguments and predicates as nodes and semantic role labels as edges. By applying a graph attention network, we can obtain syntactically and semantically augmented word representation, which can be integrated using an attentive integration layer and gating mechanism. Experiments on the OntoNotes 5.0 benchmark show the effectiveness of our proposed model.[1]

1 Introduction

Coreference resolution is a core task in NLP, which aims to identify all *mentions* that refer to the same entity. Coreference encodes rich semantic information which has been successfully applied to improve many downstream NLP tasks (Luan et al., 2019; Wadden et al., 2019; Dasigi et al., 2019; Stojanovski and Fraser, 2018).

Impressive progress has been made in recent years since the introduction of the first end-to-end neural coreference resolution model (Lee et al., 2017) by utilising contextualized embeddings from large pretrained language models (Joshi et al., 2019, 2020; Kantor and Globerson, 2019; Wu et al., 2020) such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019). Rich language knowledge encoded in these pretrained models has largely alleviated the need for syntactic and semantic features. However, such information has been shown to benefit BERT based models on other tasks (Nie et al., 2020a; Wang et al., 2020; Pouran Ben Veyseh et al., 2020). Therefore, we believe such information could also benefit the coreference resolution task.

In this paper, we propose a neural coreference resolution model based on Joshi et al. (2019), which we extend by incorporating external syntactic and semantic information. For syntactic information, we use dependency trees to capture the long-term dependency exists among mentions. Kong and Jian (2019) has successfully incorporated structural information into neural models, but their model still requires the design of complex hand-engineered features. In contrast, our model is more flexible, using a graph neural network to encode syntax in the form of dependency trees. For semantic information, we adopt semantic role labelling (SRL) structures. SRL labels capture *who did what to whom* and it is effective in providing document-level event description information, which allows us to better identify the relationship between event mentions. Previous statistical coreference systems have successfully integrated such information (Ponzetto and Strube, 2006; Kong et al., 2009), but their effectiveness has not been examined in neural models.

Moreover, inspired by recent progress made in document-level relation extraction (Christopoulou et al., 2019), we encode both syntactic and semantic information in a heterogeneous graph. Nodes of different granularity are connected based on the feature structures. Node representations are updated iteratively through our defined message passing mechanism and incorporated into contextualized embeddings using an attentive integration module and gating mechanism. We conduct experiments on the OntoNotes 5.0 (Pradhan et al., 2012) benchmark, where the results show that our proposed model significantly outperforms a strong baseline.

2 Baseline Model

Our model is based on the *c2f-coref* model (Lee et al., 2018) which enumerates all text spans as

[1] https://github.com/Fantabulous-J/coref-HGAT

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1584–1591
June 6–11, 2021.

potential mentions and prunes unlikely spans aggressively. For each mention i,[2] the model learns a distribution over its possible antecedents $\mathcal{Y}(i)$:

$$P(y) = \frac{e^{s(i,y)}}{\sum_{y' \in \mathcal{Y}(i)} e^{s(i,y')}} \quad (1)$$

where the scoring function $s(i, j)$ measures how likely span i and j comprise valid mentions and corefer to one another:

$$s(i,j) = s_m(i) + s_m(j) + s_c(i,j) \quad (2)$$
$$s_m(i) = \mathbf{FFNN}_m(\mathbf{g_i}) \quad (3)$$
$$s_c(i,j) = \mathbf{FFNN}_c(\mathbf{g_i}, \mathbf{g_j}, \phi(i,j)) \quad (4)$$

where $\mathbf{g_i}$ and $\mathbf{g_j}$ are span representations formed by the concatenation of contextualized embeddings of span endpoints and head vector using attention mechanism. **FFNN** represents the feedforward layer, $\phi(i, j)$ are meta features including span distance and speaker identities, and s_m and s_c are the mention score and pairwise coreference score.

3 Proposed Model

Figure 2 shows the architecture of our proposed model, where the key components are presented in blue and orange backgrounds. Other parts follow Lee et al. (2018) (see §2) except that we use SpanBERT (Joshi et al., 2020) as the document encoder and discard the higher-order span refinement module as suggested by Xu and Choi (2020).

3.1 Node Construction

There are three types of nodes in our heterogeneous graph: token nodes (T), argument nodes (A) and predicate nodes (P). The representation of token nodes and predicate nodes is the contextualized embeddings from the SpanBERT encoder, denoted as $\mathbf{h}_w$ and $\mathbf{h}_p$ respectively. The representation of an argument node is formed by averaging the embeddings of the tokens it contains, denoted as $\mathbf{h}_a$.

3.2 Edge Construction

Edges are constructed based on feature structures. An example is shown in Figure 1.

Token-Token Edges are constructed according to dependency tree structures. Specifically, there will be a directed edge between two token nodes starting from head to dependent if they are connected, with edges being the corresponding dependency labels. A self-loop edge with *cyclic* label is also added to each node in the graph. Besides, we also link the root nodes of two adjacent sentences to allow cross-sentence interaction.

[2] i is a span with one or more tokens.

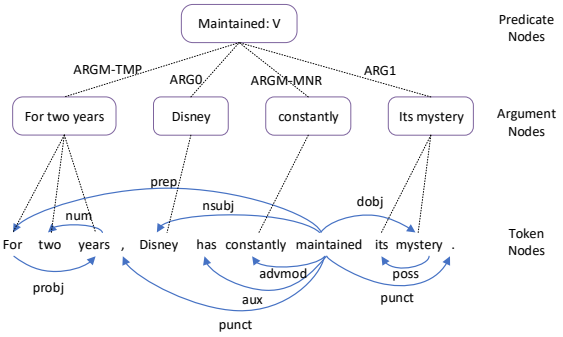

Figure 1: An example of our proposed Syntactic and Semantic based Heterogeneous Graph.

Token-Argument Argument nodes are linked to token nodes they contain. The edge is unlabelled but bidirectional to allow token-level information to augment the averaged representation of arguments and propagate semantic information back to tokens.

Predicate-Argument Argument nodes are connected to predicate nodes they belong to with edges being the corresponding SRL labels. The edge is made bidirectional to allow mutual information propagation. Predicates can be regarded as intermediate nodes to allow each argument to aggregate information from other arguments with the same predicate.

3.3 Graph Attention Layer

We use a Graph Attention Network (Veličković et al., 2018) to propagate syntactic and semantic information to basic token nodes. For a node i, the attention mechanism allows it to selectively incorporate information from its neighbour nodes:

$$\alpha_{ij} = \text{softmax}(\sigma(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i; \mathbf{W}\mathbf{h}_j; \mathbf{e}_{ij}])) \quad (5)$$
$$\mathbf{h}'_i = \|_{k=1}^{K} \text{ReLU}(\sum_j \alpha_{ij}^k \mathbf{W}^k \mathbf{h}_j) \quad (6)$$

where $\mathbf{h}_i$ and $\mathbf{h}_j$ are embeddings of node i and j, $\mathbf{a}^T$, $\mathbf{W}$ and $\mathbf{W}^k$ are trainable parameters. $\mathbf{e}_{ij}$ is the embedding of edge label type between node i and j based on graph structures, σ is the **LeakyReLU** activation function. $\|$ and $[;]$ represent the concatenation operation. Eqs. 5 and 6 are designated as an operation $\mathbf{h}'_i = \text{GAT}(\mathbf{h}_i, \mathbf{h}_j)$, where $\mathbf{h}_i$ and $\mathbf{h}_j$ are the embeddings of target and neighbour node and $\mathbf{h}'_i$ is the updated embedding of target node.

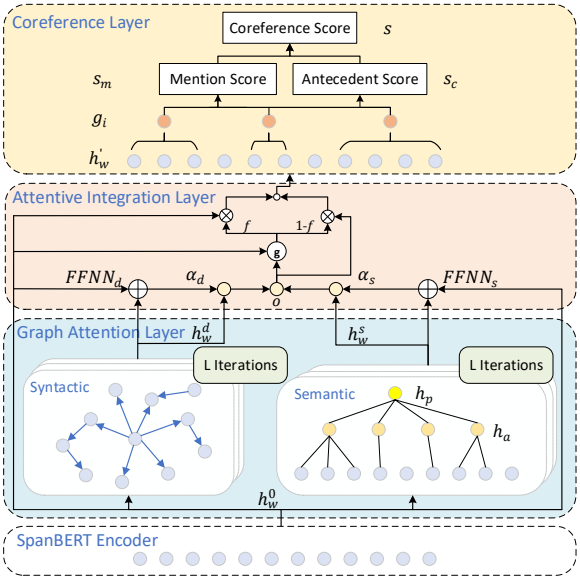

Figure 2: The architecture of our proposed model.

3.4 Message Propagation

To make each node embedding more informative, we update all nodes in the graph multiple times via our designed message passing path. First, we update token nodes using neighbour token nodes connected through dependency syntactic edges:

$$\mathbf{h}_w^l = \text{GAT}(\mathbf{h}_w^{l-1}, \mathbf{h}_w^{l-1}) \quad (7)$$

where $\mathbf{h}_w^{l-1}$ is the token representation in previous layer $l-1$, $\mathbf{h}_w^l$ is the updated representation in current layer l and $\mathbf{h}_w^0$ is the SpanBERT encoding.

In parallel, we update the argument using the token representation; then the updated argument is used to update the predicate features; after that, the updated predicate nodes propagate information back to their connected argument nodes; finally, the updated argument nodes distribute the representation to all connected basic token nodes:

$$\mathbf{h}_a^l = \text{GAT}(\mathbf{h}_a^{l-1}, \mathbf{h}_w^{l-1}) \quad (8)$$

$$\mathbf{h}_p^l = \text{GAT}(\mathbf{h}_p^{l-1}, \mathbf{h}_a^l) \quad (9)$$

$$\mathbf{h}_a^l = \text{GAT}(\mathbf{h}_a^l, \mathbf{h}_p^l) \quad (10)$$

$$\mathbf{h}_w^l = \text{GAT}(\mathbf{h}_w^{l-1}, \mathbf{h}_a^l) \quad (11)$$

After L iterations, we can get the final syntax and semantics-enhanced token representation, which can be denoted as $\mathbf{h}_w^d$ and $\mathbf{h}_w^s$, respectively.

3.5 Attentive Integration Layer

Since attention mechanisms are effective in choosing the most relevant information (Nie et al., 2020a,b), we use an attentive integration layer to selectively incorporate the syntactic and semantic information. For each type of information $\mathbf{h}_w^c \in \{\mathbf{h}_w^d, \mathbf{h}_w^s\}$, we concatenate it with initial token representation $\mathbf{h}_w^0$ and use the concatenation to compute the importance score of $\mathbf{h}_w^c$ to $\mathbf{h}_w^0$:

$$\alpha_c = \text{softmax}(\mathbf{FFNN}_c([\mathbf{h}_w^0; \mathbf{h}_w^c])) \quad (12)$$

where $\mathbf{FFNN}_c$ is a one-layer feedforward network with sigmoid activation function for information type c (either Dep or SRL). After obtaining the valid attention weights using softmax function, we could compute the weighted average sum of both syntactic and semantic information:

$$\mathbf{o} = \sum_{c \in \{d,s\}} \alpha_c \mathbf{h}_w^c \quad (13)$$

Since the extra syntactic and semantic information is not always useful, we use a gate to leverage such information dynamically:

$$\mathbf{f} = \sigma(\mathbf{W}_g \cdot [\mathbf{h}_w^0; \mathbf{o}] + \mathbf{b}_g) \quad (14)$$

$$\mathbf{h}'_w = \mathbf{f} \odot \mathbf{h}_w^0 + (1 - \mathbf{f}) \odot \mathbf{o} \quad (15)$$

where $\mathbf{W}_g$ and $\mathbf{b}_g$ are trainable parameters, $\odot$ represents element-wise multiplication and σ is the logistic sigmoid function.

Finally, the augmented token representation $\mathbf{h}'_w$ can be used to form span representation and compute pairwise coreference score as in Section 2.

4 Experiments

Dataset We evaluate our model on the English OnotoNotes 5.0 benchmark (Pradhan et al., 2012), which consists of 2802, 343 and 348 documents in the training, development and test data sets.

Implementation Details We reimplement the *c2f-coref+SpanBERT*[3] baseline using PyTorch and use the *Independent* setup for long documents. For graph encoders, the number of heads of syntactic and semantic sub-graphs is 4 and 8 for base and large model, respectively. We set the size of edge label embeddings to 300 and use 2 GAT layers for both sub-graphs. More details are in Appendix A.

Results The main evaluation is the average F1 of three metrics – MUC, B^3 and CEAFϕ_4 on the test set using the official CoNLL-2012 evaluation scripts.[4] Table 1 shows the results of coref-HGAT

[3]https://github.com/mandarjoshi90/coref
[4]http://conll.cemantix.org/2012/software.html

	MUC			B^3			$CEAF_{\phi_4}$			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
e2e-coref (Lee et al., 2017)	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
c2e-coref (Lee et al., 2018)	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
EE (Kantor and Globerson, 2019)	82.6	84.1	83.4	73.3	76.2	74.7	72.4	71.1	71.8	76.6
SpanBERT-base (Joshi et al., 2020)	84.3	83.1	83.7	76.2	75.3	75.8	74.6	71.2	72.9	77.4
Our baseline + SpanBERT-base*†	83.6	83.9	83.7	75.1	76.5	75.8	74.2	71.6	72.9	77.5 ($\pm$0.1)
coref-HGAT + SpanBERT-base†	**85.1**	**84.5**	**84.8**	**77.4**	**77.2**	**77.3**	**75.5**	**73.3**	**74.4**	**78.8** ($\pm$0.1)
SpanBERT-large (Joshi et al., 2020)	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	79.6
Our baseline + SpanBERT-large*†	85.7	85.6	85.6	78.5	78.7	78.6	76.5	75.0	75.7	80.0 ($\pm$0.1)
coref-HGAT + SpanBERT-large†	**86.8**	**86.3**	**86.5**	**80.0**	**79.7**	**79.8**	**78.0**	**75.9**	**76.9**	**81.1** ($\pm$0.2)
CorefQA (Wu et al., 2020)	88.6	87.4	88.0	82.4	82.0	82.2	79.9	78.3	79.1	83.1

Table 1: The results on the test set of the OntoNotes English shared task compared with previous systems. The main evaluate metric is the averaged F1 of MUC, B^3 and CEAFϕ_4. $*$ indicates our reimplemented baseline. $\dagger$ indicates average performance over 5 runs using different random seeds.

	Avg. F1	$\Delta F1$
Baseline	77.5	-
+ Dep	78.5	+1.0
+ SRL	78.4	+0.9
+ Dep & SRL	78.8	+1.3
GAT Layer = 1	78.5	-0.3
GAT Layer = 2	78.8	-
GAT Layer = 3	78.6	-0.2

Table 2: The Avg. F1 of coref-HGAT Base model by adding different features and stacking different number of GAT layers on the test set.

	Dep	SRL	F1	$+\Delta F1$
Baseline	-	-	77.5	-
	Stanford	CoNLL05-SRL	78.1	+0.6
	Stanford	CoNLL12-SRL	78.2	+0.7
	Biaffine	CoNLL05-SRL	78.2	+0.7
	Biaffine	CoNLL12-SRL	78.4	+0.9

Table 3: The Avg. F1 of coref-HGAT+Base model with predicted features against the baseline.

+SpanBERT-base and large model compared with previous work. Our model consistently outperforms the SpanBERT baseline (Joshi et al., 2020) on all three metrics with an improvement of 1.4% and 1.5% on Avg. F1 score respectively, as well as our reimplemented baseline (+1.3% and +1.1%), which is a substantial improvement by considering the difficulty of this task. This demonstrates the effectiveness of our heterogeneous graph-based method in leveraging syntactic and semantic features and such features are indeed useful in neural methods. Note that we also show the current state-of-the-art CorefQA model (Wu et al., 2020), which uses span-prediction paradigm to compute pairwise coreference scores. The model is compatible with our method, i.e. adding our proposed graph attention and attentive integration layer on top of their document encoder with minor modification. The reason why we did not use it as a start baseline is due to hardware limitations since it requires 128G GPU memory for training.

Ablation Study We perform ablation study on the test set to investigate the contribution of different features in our model, with results shown in Table 2. We can see that both dependency features and SRL labels individually contribute to the success of our final model with minor difference (+1.0% and 0.9%), and the gains are complementary to each other.

Effect of #Graph Layers From Table 2, we can see that both using one layer and three layers hurt model performance. This indicates that first-order information is not effective in capturing long-range dependencies while third-order information may cause overfitting due to too much model capacity.

Effect of Feature Quality To evaluate how the quality of features will affect the performance, we use the biaffine dependency parser (Dozat and Manning, 2017) and SRL parser (Shi and Lin, 2019) (denoted as *CoNLL12-SRL*) implemented by AllenNLP (Gardner et al., 2018) as well as the Stanford Parser (Chen and Manning, 2014) to extract features. The biaffine parser has roughly 3% LAS improvements compared to the Stanford CoreNLP parser on Penn Treebank. Moreover, in order to

Doc length	#Docs	Baseline	Ours	$+\Delta F1$
0 – 128	57	82.9	85.4	+2.5
129 – 256	73	81.8	83.1	+1.3
257 – 512	78	82.2	83.2	+1.0
512 – 768	71	77.7	78.2	+0.5
769 – 1152	52	76.8	78.6	+1.8
1153+	12	67.5	70.3	+2.8
All	343	77.8	79.2	+1.4

Table 4: The Avg. F1 on the development set of the SpanBERT-base model and our core-HGAT+Base model, broken down by document length following Xia et al. (2020).

evaluate the impact of different SRL parsers, we also implemented the same model from Shi and Lin (2019) but trained on the CoNLL 2005 dataset (Carreras and Màrquez, 2004) (denoted as *CoNLL05-SRL*), which achieves an F1 of 81.9% on the out-of-domain setting. From Table 3, we observe that better parsers and parsers trained in closer domains result in higher Avg. F1 score, with improvements of up to 0.9%. Meanwhile, although our model suffers a performance drop from imperfect features, it can still achieve robust performance, outperforming the baseline with at least 0.6% improvement. Overall, high-quality features are important to good performance of the proposed model.

Document Length In Table 4, we show the performance of our model against the baseline on the development set as a function of document lengths. As expected, our model consistently outperforms the baseline model on all document sizes, especially for documents with length larger than 765 tokens. This demonstrates that the incorporated external syntax and semantics are beneficial for modelling longer dependencies. However, our model has similar pattern as the baseline model, performing distinctly worse as document length increases. This shows that the sentence-level syntax and semantics used in this work are not sufficient enough to tackle the deficiency of modelling long-range dependency. One possible solution is to leverage document-level features such as hierarchical discourse structures.

5 Related Work

Graph Neural Networks (GNN) have long been used for integrating external features of graph structures into a range of NLP tasks, including semantic role labelling (Marcheggiani and Titov, 2017) and machine translation (Bastings et al., 2017). However, the application of GNN on coreference resolution task is less explored. Xu and Yang (2019) adopted dependency syntax to improve gendered pronoun resolution. However, they did not evaluate their model on larger datasets and identify whether syntax features are still useful for common coreference resolution. In this paper, we not only utilise syntax but also semantic features, and we show both of them contribute to significant improvement over a strong baseline on a large standard dataset.

There are many GNN variants. Graph Convolutional Network (GCN) (Kipf and Welling, 2017) is the most widely-used one and has been shown to benefit a number of NLP tasks. However, it lacks the ability of modeling different edge labels including directions and edge types. Although Relational Graph Convolutional Network (RGCN) (Schlichtkrull et al., 2017) was proposed to tackle this problem, the way of representing edge information as label-wise parameters makes it suffer from over-parameteration problem even for small sized label vocabularies. In this work, we use a graph encoder improved based on Graph Attention Network (GAT) (Veličković et al., 2018) to better capture structural syntax and semantics, as GAT is able to model different types of edges with few parameters.

6 Conclusion

In this paper, we propose a heterogeneous-graph based model to enhance coreference resolution by effectively leveraging dependency tree structures and SRL semantic features. Particularly, nodes of different granularity in the graph propagate and aggregate information to and from neighbour nodes to obtain both syntactically and semantically augmented representation. Moreover, an attention-based mechanism is used to dynamically aggregate such augmented information. Experiments on the OntoNotes 5.0 benchmark confirm the effectiveness of our proposed model with significant improvement achieved against the strong baseline. Future work will focus on applying other features, such as constituent parsing trees and WordNet.

Acknowledgements

We thank the anonymous reviewers for their helpful feedback. This research was undertaken using the LIEF HPC-GPGPU Facility hosted at the University of Melbourne. This Facility was established with the assistance of LIEF Grant LE170100200.

References

Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967, Copenhagen, Denmark. Association for Computational Linguistics.

Xavier Carreras and Lluís Màrquez. 2004. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 89–97, Boston, Massachusetts, USA. Association for Computational Linguistics.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4925–4936.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5925–5932.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5803–5808.

Ben Kantor and Amir Globerson. 2019. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Fang Kong and Fu Jian. 2019. Incorporating structural information for better coreference resolution. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5039–5045. International Joint Conferences on Artificial Intelligence Organization.

Fang Kong, GuoDong Zhou, and Qiaoming Zhu. 2009. Employing the centering theory in pronoun resolution from the semantic perspective. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 987–996.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197. Association for Computational Linguistics.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Asso-*

ciation for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 3036–3046.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Yuyang Nie, Yuanhe Tian, Yan Song, Xiang Ao, and Xiang Wan. 2020a. Improving named entity recognition with attentive ensemble of syntactic information. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4231–4245.

Yuyang Nie, Yuanhe Tian, Xiang Wan, Yan Song, and Bo Dai. 2020b. Named entity recognition for social media texts with semantic augmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1383–1391.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Simone Paolo Ponzetto and Michael Strube. 2006. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 192–199.

Amir Pouran Ben Veyseh, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Graph transformer networks with syntactic and semantic structures for event argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3651–3661.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks.

Peng Shi and Jimmy Lin. 2019. Simple BERT models for relation extraction and semantic role labeling. *CoRR*, abs/1904.05255.

Dario Stojanovski and Alexander Fraser. 2018. Coreference and coherence in neural machine translation: A study using oracle experiments. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 49–60.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. CorefQA: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6953–6963.

Patrick Xia, João Sedoc, and Benjamin Van Durme. 2020. Incremental neural coreference resolution in constant memory. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8617–8624, Online. Association for Computational Linguistics.

Liyan Xu and Jinho D. Choi. 2020. Revealing the myth of higher-order inference in coreference resolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8527–8533.

Yinchuan Xu and Junlin Yang. 2019. Look again at the syntax: Relational graph convolutional network for gendered ambiguous pronoun resolution. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 96–101, Florence, Italy. Association for Computational Linguistics.

A Implementation Details

We utilise the Adam Optimizer (Kingma and Ba, 2015) with a gradient clipping of 1.0 and a batch size of 1 (single document) for both base and large models. SpanBERT-base and large models are fintuned using learning rates of 2×10^{-5} and 1×10^{-5}, with a warmup scheduler in the first 10% training steps. We use learning rates of 3×10^{-4} and 5×10^{-4} for task-related parameters with linear decay decreasing to 0. The training of base model is conducted on a single Nvidia Telsa V100 GPU with 16G memory while training large model requires 32G memory.

Gold features annotated on the OntoNotes 5.0 dataset are used in the experiment. We use Stanford CoreNLP toolkit (Manning et al., 2014) to convert the annotated constituent trees into Stanford dependency trees (de Marneffe and Manning, 2008). SRL labels are organized in the form of triples: (p, a, l), which refers to predicate, argument and label, respectively.