

Joint Semantic Analysis with Document-Level Cross-Task Coherence Rewards

Rahul Aralrikatte, Mostafa Abdou, Heather C Lent,
Daniel Hershcovich and Anders Søgaard

University of Copenhagen
{rahul, abdou, hcl, dh, soegaard}@di.ku.dk

Abstract

Coreference resolution and semantic role labeling are NLP tasks that capture different aspects of semantics, indicating respectively, which expressions refer to the same entity, and what semantic roles expressions serve in the sentence. However, they are often closely interdependent, and both generally necessitate natural language understanding. Do they form a coherent abstract representation of documents? We present a neural network architecture for joint coreference resolution and semantic role labeling for English, and train graph neural networks to model the *coherence* of the combined shallow semantic graph. Using the resulting coherence score as a reward for our joint semantic analyzer, we use reinforcement learning to encourage global coherence over the document and between semantic annotations. This leads to improvements on both tasks in multiple datasets from different domains, and across a range of encoders of different expressivity, calling, we believe, for a more holistic approach to semantics in NLP.

Introduction

Coreference resolution and semantic role labeling (SRL) contribute in complimentary ways to forming coherent discourse representations. SRL establishes predicate-argument relations between expressions, and coreference resolution determines what entities these expressions refer to. While often treated separately (He et al. 2017, 2018; Lee et al. 2017; Lee, He, and Zettlemoyer 2018; Joshi et al. 2019), some frameworks consider coreference and semantic roles part of a more holistic meaning representation (Shibata and Kurohashi 2018). For example, the Groningen Meaning Bank (Bos et al. 2017) annotates documents with discourse representation structures (Kamp and Reyle 2013), which subsume both levels of analysis; the same holds for other meaning representation frameworks, such as UCCA (Abend and Rappoport 2013; Prange, Schneider, and Abend 2019) and AMR (Banarescu et al. 2013; O’Gorman et al. 2018). However, these frameworks do not offer the simplicity of SRL and coreference annotation, and perhaps consequently require more effort to annotate, and do not have the same amounts of training data (Abend and Rappoport 2017). Furthermore, comprehensive meaning representation *parsing* approaches (Liu, Cohen, and Lapata 2018; Hershcovich, Abend, and

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

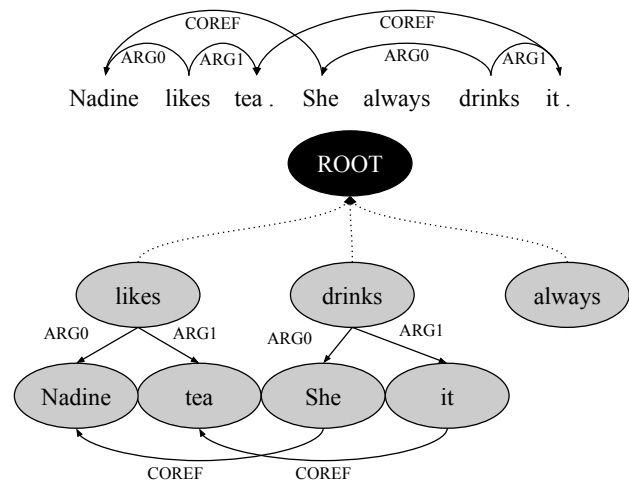


Figure 1: Example coreference and semantic role annotation for a two-sentence document. Top: the original annotation shown as dependencies. Bottom: shallow semantic graph (SSG), where sub-graph heads are connected (with dotted lines) to a dummy root node.

Rappoport 2017; Cai and Lam 2020) tend to be more complex than the sequence tagging or span-based models often used for coreference resolution and SRL, often referred to as *shallow semantic parsing*.

In this paper, we investigate a “minimal” approach to discourse-level semantic parsing, combining coreference and semantic roles in *shallow semantic graphs* (SSGs) that can be seen as a simple, yet rich, discourse-level meaning representations. Consider the two sentences shown in Figure 1, augmented with a (partial) annotation of coreference and semantic roles. A coreference resolver is expected to resolve *Nadine* as an antecedent of *she*, and *tea* as an antecedent of *it*, since these mentions refer to the same entities. A semantic role labeler is expected to detect that these entities are arguments of the predicates *like* and *drink*. The overall semantic analysis corresponds to a coherent and common situation, where someone likes something and consumes it—a very plausible interpretation. This paper presents a model that scores the plausibility or *coherence* of an interpretation based

on merged SRL and coreference graphs, or SSGs. While Figure 1 is a simple example that existing SRL and coreference systems will likely handle well, we explore whether such systems in general benefit from feedback from a model that rewards the coherence of their output.

Contributions We jointly model coreference resolution and SRL to form discourse-level semantic structures, or SSGs (§). We explicitly model their coherence, presenting a reinforcement learning architecture for semi-supervised fine-tuning of coreference resolvers and semantic role labelers with coherence rewards on unlabeled data (§), improving both coreference resolution and SRL. We present experiments across six encoders of different complexities, six different coreference resolution datasets, and four different SRL datasets (§), showing improvements across all encoders for coreference resolution, and on 4/6 for SRL, for single-task setups; and similar improvements in multi-task setups, where encoder parameters are shared across the two tasks (§). Finally, we analyze the results (§), showing that our fine-tuning setup is particularly beneficial for smaller documents while being on-par with strong baselines on larger documents and that the majority of the remaining coreference errors occur when the antecedent is a pronoun.

Joint Coreference Resolution and SRL

We build baseline single-task and multi-task *supervised models* for coreference resolution and SRL. The overall model architecture is illustrated in Figure 2 (bottom half; till the coreference clusters and SRL tags are generated). In the multi-task setup only the contextualizing encoder is shared. In the single-task setup no parameters are shared.

Coreference Resolver

The coreference model is based on the architecture presented in Lee et al. (2017). Each token’s embedding is obtained using a contextualizing encoder. Using a *span encoder*, the token embeddings are combined into span representations $s(i, j)$, where i and j are the start and end indices in the document. Each span is represented as the concatenation of: (i) its first and last token embeddings, and (ii) an attention-based aggregation of embeddings of all tokens in the span. These span representations are pruned with a *mention scorer*, which outputs the probability of $s(i, j)$ being a coreferent mention. Next, the mention representations are paired together and scored again with a *pair scorer*, which predicts the probability of the mentions referring to each other. Coreferent mentions are collected to form clusters. This architecture is combined with pre-trained language models in Lee, He, and Zettlemoyer (2018) and Joshi et al. (2019) to get state-of-the-art results.

Semantic Role Labeler

The SRL tagger is based on the architecture presented in He et al. (2017). The model uses the contextualizing encoder to embed tokens which are concatenated with a binary indicator to identify whether the token is a verb or not. These token representations are presented to a *argument classifier* for BIO sequence tagging. The current state-of-the-art (He et al. 2018)

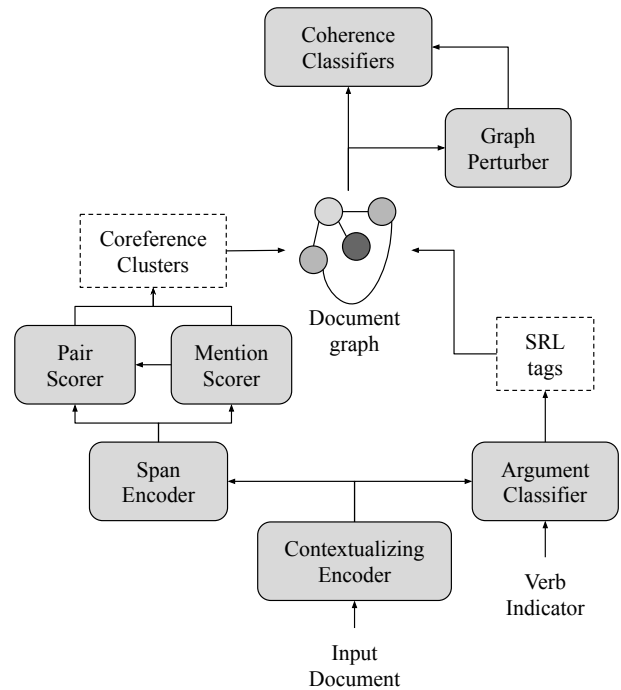


Figure 2: Joint coreference resolution and SRL (bottom half) with a coherence objective (top half). The contextualizing encoder is shared in the multi-task setup, and separate in the single-task one. Predictions from the coreference and SRL models are combined to a document-level SSG, which is scored by coherence classifiers to reward the models.

uses an architecture similar to that of Lee et al. (2017), where it jointly predicts both arguments and predicates.

Contextualizing Encoder

In all setups, we experiment with (i) an LSTM + CNN encoder, and (ii) five BERT (Devlin et al. 2019) encoders of different sizes. In the LSTM + CNN encoder, a bi-LSTM contextualizes words embedded with GloVe (Pennington, Socher, and Manning 2014) and a CNN encodes individual characters. The final representation is the concatenation of the two. For the BERT encoders, we experiment with different encoder sizes as shown in Table 2, using each token’s wordpiece embeddings. Encoder hyperparameters are given in §.

Semi-Supervised Fine-Tuning

In the semi-supervised stage of training, classifiers trained on SSGs created from labeled data (Figure 1) are used to fine-tune the supervised models on unlabeled data by reinforcement learning. For each unlabeled document, we use the predicted annotations of the supervised models to build an SSG consisting of SRL predicates and arguments, with links between coreferent mentions. Edge labels are used to distinguish between SRL and coreference edges. These graphs are scored by *graph classifiers* (§), trained using graph perturbations (§) to model semantic coherence. The confidence value

is used as a reward to fine-tune the supervised models using policy gradient (§).

Coherence Classifiers

We use a graph convolution network (GCN; Kipf and Welling 2017) to construct continuous representations of the SSGs, where a node representation is composed via a learnt weighted sum of neighboring nodes. Since nodes correspond to text spans, to initialize their representations, we use the supervised model’s span encoder. To get the final graph encoding, all the node representations are averaged and compressed using the logistic function as shown in Equation 1.

$$\text{graph}_{enc} = \sigma \left(\frac{1}{N} \sum_{i=1}^N \text{node}_{enc}^i \right) \quad (1)$$

The GCN parameters are pre-trained using deep graph infomax (DGI; Veličković et al. 2018), which relies on graph perturbations to learn a task-independent representation. We contrastively train the GCN encoder on gold and *perturbed* graphs, which are generated by randomly perturbing the gold graphs (§). We then use the same perturbations to train a logistic regression classifier, with the GCN outputs as features, to discriminate gold graphs from perturbed graphs. As shown in §, the trained classifiers are almost perfectly accurate on an unseen development set.

The process for training the coherence classifiers is shown in Algorithm 1. First an SSG $g \in \mathbb{G}$ is built for each labeled document. Then for each type of perturbation $p \in P$, we train one classifier as follows: (i) perturb g to get g_p using perturbation p . We use a decay factor $d \in \{0, 1\}$ to decide the probability of perturbing a sentence in the document. We start with $d = 0.8$ and decay it till $d = 0.1$, (ii) once we have a list of perturbed graphs \mathbb{G}_p , we train the GCN using DGI, which uses a contrastive loss to learn graph representations such that each pair (g, g_p) is as different to each other as possible, (iii) we use the GCN to get the final representations of graphs in \mathbb{G} and \mathbb{G}_p and create a training dataset consisting of the following (graph, label) pairs: $\{(g, 1) : g \in \mathbb{G}\} \cup \{(g_p, 0) : g_p \in \mathbb{G}_p\}$, and (iv) we train a logistic regression classifier.

Graph Perturbations

To train the GCN with DGI, we perturb the gold graphs to reflect the statistics of errors made by the supervised models we want to fine-tune. In general, perturbations are sampled from the following operations: (i) randomly removing edges, (ii) randomly adding edges between existing nodes with a random label, or (iii) randomly adding nodes with a span that is a constituent in the sentence, and a random edge to another existing node. We arbitrarily choose to sample SRL and coreference perturbations with a 3-to-1 ratio.

For SRL perturbations, we rely on the error analysis made by He et al. (2017), whose SRL model is the basis for ours: 29.3% of errors correspond to incorrect argument labels; 4.5% to moved unique arguments; 10.6% to split arguments; 14.7% to merged arguments; 18% to incorrect boundaries; 7.4% to superfluous arguments; and 11% to missed arguments. Consequently, we sample perturbations proportionally to the corresponding error’s frequency. We further use He

Algorithm 1 Training Coherence Classifiers

Require: \mathbb{G} : List of SSGs
Require: P : List of perturbations to perform
Require: d : Decay factor
Initialize $\text{clfs} = \emptyset$
for p in P **do**
 for epoch = 1, ..., N **do**
 Initialize $\mathbb{G}_p = \emptyset$
 for g in \mathbb{G} **do**
 $g_p = p(g, d)$
 $\mathbb{G}_p.add(g_p)$
 end for
 encoder = DGI(\mathbb{G}, \mathbb{G}_p)
 $d = \text{decay}(d)$
 end for
 $\text{data}_+ = (\text{encoder}(\mathbb{G}), 1)$
 $\text{data}_- = (\text{encoder}(\mathbb{G}_p), 0)$
 $\text{clf}_p = \text{logistic}(\text{data}_+, \text{data}_-)$
 $\text{clfs.add}(\text{clf}_p)$
end for
return clfs

et al. (2017)’s observed confusion matrix of predicted and gold argument labels, sampling replacement labels accordingly. For *coreference* perturbations, we add a random edge between existing nodes or remove an edge, with uniform probability.

We train one classifier to identify each type of perturbation, resulting in nine different classifiers (seven for SRL and two for coreference; an example for one of each is illustrated in Figure 3). The final confidence for a graph is the average of the individual classifier confidence scores.

Model Fine-Tuning

Finally, we use the learned classifiers to fine-tune the underlying coreference resolver and semantic role labeler; using plain text from summary paragraphs of Wikipedia articles, we apply the supervised models to sample an SSG. Using the coherence classifiers’ confidence score as a reward, we train the models with policy gradient.

During policy gradient, we consider the selection of SSG edges as actions. More concretely, for coreference resolution, picking the antecedent to each mention is considered an action. Therefore from Figure 1, assuming the model found four mentions (‘Nadine’, ‘tea’, ‘She’, and ‘it’), it takes four actions (connecting ‘Nadine $\rightarrow \phi$ ’, ‘tea $\rightarrow \phi$ ’, ‘she \rightarrow Nadine’, ‘it \rightarrow tea’).¹ For SRL, assigning a label to a token is considered as an action. Therefore the model has to perform nine actions (one for each token) to label Figure 1.

In this work, we assume that all actions are equally good and reward them uniformly. Assigning rewards to individual actions would probably yield better results but is non-trivial and left for future exploration.

¹ ϕ indicates no antecedent

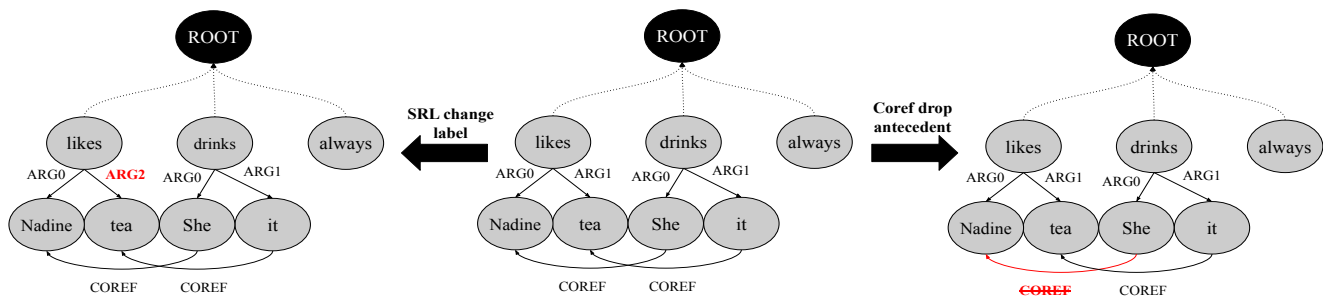


Figure 3: Examples for graph perturbations, starting from the SSG in Figure 1 (center). An ‘SRL change label’ perturbation is applied to generate a graph (left), where ARG1 is changed to ARG2. A ‘Coref drop antecedent’ perturbation is applied to generate a graph (right) where a COREF edge is deleted.

Experiments

In this section, we briefly describe the datasets used to train and evaluate our models before moving on to the experimental setup. We then provide implementation details for each stage of the training process and finally present the results of our experiments.

Datasets

For supervised training, we use data from the CoNLL-2012 shared task (Pradhan et al. 2012), which contains data from OntoNotes 5.0² with annotations for both coreference resolution and semantic role labeling.

As additional out-of-domain (OOD) development and test data for coreference resolution, we use (i) PreCo (Chen et al. 2018), which contains web-crawled documents and data from the RACE dataset (Lai et al. 2017); (ii) Phrase Detectives (Poesio et al. 2013), which contains two evaluation sets, one sampled from Wikipedia and the other from the Gutenberg project; (iii) WikiCoref (Ghaddar and Langlais 2016), which contains long form documents from the English Wikipedia; and (iv) WinoBias (Zhao et al. 2018), which is focused on gender bias with Winograd-schema style sentences, authored manually.

For SRL, we additionally use (i) the CoNLL-2005 shared task data (Carreras and Màrquez 2005), which contains two evaluation sets: the in-domain WSJ set and the OOD Brown set; and (ii) English Web Treebank (Silveira et al. 2014)³, which contains weblogs, newsgroups, email, question-answers and review text.

Experimental Setup

We first train the coreference and SRL models (§) using supervised learning, and the coherence classifiers on gold graphs and their perturbations. Both are trained on the CoNLL-2012 training set. We then fine-tune the models by semi-supervised learning (§), with the summary paragraphs of 10,000 randomly sampled English Wikipedia articles.⁴ We test our mod-

Hyperparameters	Lee et al. (2018)	Joshi et al. (2019)	Ours
max. span width	30	30	10
cxt. enc. (layers/dims)	3/1024	24/1024	12/768*
span enc. (layers/dims)	3/400	-	1/400
pruner (layers/dims)	2/150	1/1000	1/150
top span ratio	0.4	0.4	0.3
max antecedents	250	50	100
course to fine inference	True	True	False

Table 1: Comparison of hyperparameters between state-of-the-art and our coreference models. *This value is for BERT-Base. See Table 2 for other sizes.

els across six domains for coreference resolution, and four domains for SRL, using in-domain evaluation data.

Implementation Details

Since the goal of this work is not to surpass the state of the art, but to demonstrate that discourse-level coherence can be used to improve shallow semantic analysis, and due to memory and compute constraints, we use smaller versions of the best performing architectures in the literature as baselines.

Coreference Model

We use the same architecture that state-of-the-art coreference systems like Lee et al. (2017); Lee, He, and Zettlemoyer (2018) and Joshi et al. (2019) use, but with lesser capacity. A comparison of the important hyperparameters that vary between our model and the current state-of-the-art is shown in Table 1.

SRL Model

He et al. (2017) use 8 LSTM layers with highway connections and recurrent dropout. We replace this encoder with each of our contextualizing encoder configurations. Following He et al. (2017), we also use constrained decoding to produce only valid BIO tags as output.

²<https://catalog.ldc.upenn.edu/LDC2013T19>

³<https://catalog.ldc.upenn.edu/LDC2017T15>

⁴<https://www.wikipedia.org>, dump from March 4, 2019.

Encoder	# layers	dim
LSTM + CNN	1	500
BERT-Tiny	2	128
BERT-Mini	4	256
BERT-Small	4	512
BERT-Medium	8	512
BERT-Base	12	768

Table 2: Number of layers and the output dimension of our contextualizing encoders.

Contextualizing Encoders

For the LSTM + CNN encoder, 300-dimensional GloVe embeddings (Pennington, Socher, and Manning 2014) are fed into a bi-LSTM with a hidden size of 200, to get a 400-dimensional word representation. We concatenate this with 100-dimensional character embeddings obtained from a CNN character encoder with a filter size of 5. The other five encoders are based on the standard BERT recipe (Turc et al. 2019), and their sizes can be seen in Table 2.

Supervised Training

For training both single-task and multi-task models, we use the Adam optimizer (Kingma and Ba 2015) with a weight decay of 0.01 and initial learning rate of 10^{-3} . For BERT parameters, the learning rate is lowered to 10^{-5} . We reduce the learning rates by a factor of 2 if the evaluation on the development sets does not improve after every other epoch. The training is stopped either after 100 epochs, or when the minimum learning rate of 10^{-7} is reached. In the multi-task setup, we sample a batch from each task with a frequency proportional to the dataset size of that task. All experiments are run on a single GPU with 16GB memory. The hyperparameters were manually selected to accommodate for training time and resource limitation, and were not tuned based on model evaluation.

Coherence Classifiers

The GCN encoder used to encode the SSGs has 512 hidden channels and is trained with Adam for 10 epochs. We use a 20-dimensional embedding to represent the type of node and a binary indicator to represent the edge type.

Finetuning

The supervised models are fine-tuned for 10 epochs with the same optimizer configuration. Only the learning rate is changed to $3 \cdot 10^{-4}$. Hill climbing is used during policy gradient, i.e., if fine-tuning on a batch of Wikipedia documents does not yield an improvement, the parameters are reset to their previous best state.

In the multi-task setup, the coreference resolution and SRL sub-models are fine-tuned separately. This is because we do not want to sample actions for both tasks as it makes the constructed SSG more noisy. For constructing the SSGs in the single-task setup, we use the best performing SRL model for fine-tuning the coreference resolution model, and the best

	Perturbation type	Accuracy (%)
SRL	change label	98.98
	move argument	99.88
	split spans	99.72
	merge spans	99.29
	change boundary	98.96
	add argument	99.22
	drop argument	100.00
Coref	add antecedent	99.10
	drop antecedent	100.00

Table 3: Graph classifier development accuracy.

performing coreference resolution model for fine-tuning the SRL model.

Results

Coreference Resolution and SRL

The mean F_1 over MUC, CEAF $_{\phi_4}$, and B^3 scores averaged across the six test sets for coreference resolution and four test sets for SRL (including in-domain and out-of-domain), for each of the six encoder configurations, is presented in Table 4. The individual results for each dataset is presented in Tables 5 and 7 for single-task models, and in Tables 6 and 8 for multi-task models respectively.

We see substantial improvements from coherence fine-tuning across the board for all coreference tasks. Results for single-task SRL improves in all settings except for BERT-mini and BERT-medium encoders. In the multi-task setting for SRL, we see consistent improvements with two exceptions: the results for LSTM + CNN and BERT-base. Coreference resolution generally improves more than for SRL.

Coherence Classifiers

The accuracy of the nine coherence classifiers (§) on the CoNLL-2012 development set is shown in Table 3, showing that the classifiers can almost perfectly detect perturbed graphs, and explaining their effectiveness at providing a reward signal to the models. While it could be argued that this indicates that the perturbations are too easy to detect, observing the perturbed graphs (exemplified in Figure 3) leads to the impression that they require sensitivity to distinctions that are important for correct coreference resolution, SRL and the coherence between them. Indeed, the rewards lead to improvements in each of the tasks.

Error Analysis

By analysing the results of the fine-tuned models on all datasets (Table 4), we make the following observations:⁵

Document length Fine-tuning leads to larger improvements on smaller documents (see Figure 4). This is likely because the unlabeled data we use for fine-tuning consists

⁵Unless mentioned otherwise, all analysis is carried out on the single-task BERT-Base model.

Encoder	Single-Task				Multi-Task			
	Coreference		SRL		Coreference		SRL	
	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours
LSTM + CNN	49.01	49.40	67.63	67.74	48.65	49.60	67.28	67.05
BERT-Tiny	49.70	50.95	56.87	57.08	45.65	51.17	56.65	56.85
BERT-Mini	52.61	52.88	70.51	70.48	50.14	53.02	71.10	71.13
BERT-Small	52.76	53.90	74.26	74.48	51.26	53.73	74.72	74.77
BERT-Medium	55.67	56.19	75.62	75.57	51.48	55.52	77.89	78.01
BERT-Base	57.78	58.18	79.46	79.52	56.40	57.55	80.25	80.19

Table 4: COREFERENCE RESOLUTION and SEMANTIC ROLE LABELING results of single-task and multi-task models. ‘Baseline’ and ‘Ours’ represent the supervised baseline and coherence fine-tuned models respectively. The numbers are the mean of MUC, B³ and CEAF_{φ₄}F₁ scores averaged over six (four) coreference (SRL) datasets.

of short paragraphs. While using longer documents for fine-tuning was not possible due to memory constraints, we expect that this will increase the model’s sensitivity to long-distance inter-dependencies, and further improve its performance on these documents.

Coreference resolution vs. SRL In general, SRL sees smaller improvements from fine-tuning with policy gradient than coreference resolvers, probably because it is harder to assign credit to specific model decisions (Langford and Zadrozny 2005). Semantic role labeling of a paragraph typically requires a much longer sequence of actions than determining coreference, leading to limited benefit from reinforcement learning. Similar results have been observed in machine translation (Choshen et al. 2020).

Precision vs. recall Precision often increases after fine-tuning whereas recall decreases. Similar effects have been reported for knowledge-base grounding of coreference resolvers (Aralikatte et al. 2019).

Encoder sizes From the results, we also see that our fine-tuning approach is robust to encoder sizes with improvements across the board. It is particularly interesting to see that the multi-task BERT-Tiny coreference models come close or even surpass the bigger BERT-Base models on datasets like PreCo and WinoBias, which contain short documents (see Table 6).

In both single-task and multi-task setups, fine-tuning helps the smaller coreference models more than the larger ones, which are already more accurate. This trend is expected as the larger models tend to be over-parameterized.

Domain adaptation We also perform an error analysis to identify the domains which are hard for our coreference models (see Figure 5). We find that our coherence fine-tuned (CO) model always performs better than or on par with the supervised baseline (SU) model, except in the case of Phrase Detectives - Gutenberg (PD-G). We postulate that the increase in PD-G errors can be attributed to the length of the documents in the dataset.⁶

⁶The average document length of PD-G is 1507.2 tokens, which is the highest among all datasets.

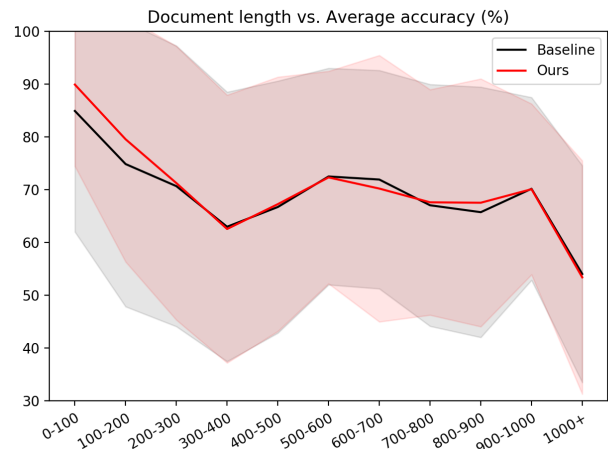


Figure 4: Percentage of correct predictions of our BERT-Base coreference model across all datasets plotted against document lengths.

Part-of-speech As seen in Figure 6, across all domains, most errors from the coherence fine-tuned system occur when the antecedent is a pronoun, except for WikiCoref, where the most errors occur when the antecedent was a multi-word expression. This trend is seen in the supervised baseline models as well.

Apart from being the most frequent among mentions, two possible reasons why pronouns could be predicted incorrectly most often are: (i) as the distance in text increases between the original antecedent and subsequent pronouns, it becomes more difficult to resolve, and (ii) as a text becomes more complex, with multiple possible antecedents to choose from, linking becomes harder. Given the increased performance of our coreference resolver from the inclusion of a coherence classifier, we hypothesize that the second problem would be easier for our system to overcome, while the first could still persist.

Span length Finally, we analyse the length of the mentions linked by our models. In general, both supervised baseline and coherence fine-tuned models perform similarly for very

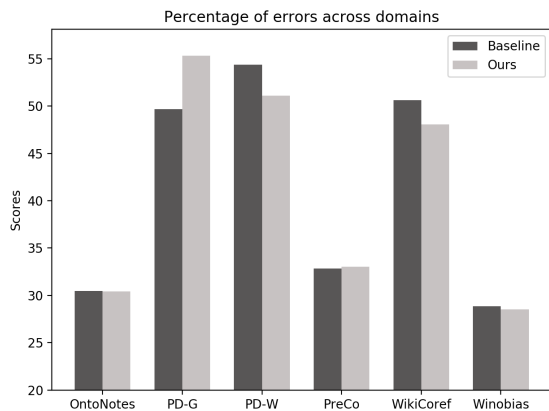


Figure 5: Percentage of errors over the total amount of predictions that our coreference system makes across each domain of the evaluation data.

short (0–3 tokens) and very long (7+ tokens) mentions. However, we see an improvement in linking accuracy of the coherence fine-tuned model when the mention length is between 3–7.

Related Work

Augmented Coreference Resolution

Previous work has augmented Coreference resolvers with syntax information (Wiseman, Rush, and Shieber 2016; Clark and Manning 2016a,b), external world knowledge (Rahman and Ng 2011; Emami et al. 2018; Aralikkatte et al. 2019) and a variety of other linguistic features (Ng 2007; Haghighi and Klein 2009; Zhang, Song, and Song 2019). Similarly, Ponzetto and Strube (2006a,b) used features from SRL and external sources for a non-neural coreference resolver.

Augmented Semantic Role Labelling

SRL systems have long utilised annotations from syntactic formalisms as an essential component (Levin 1993; Hacioglu 2004; Pradhan et al. 2005; Sutton and McCallum 2005; Panyakanok, Roth, and Yih 2008). More recently, Strubell et al. (2018) showed that it was possible to exploit information from syntactic parses for supervision of the self-attention mechanism in a fully differentiable transformer-based SRL model, surpassing the previous state-of-the-art. Xia et al. (2019) follow up on this, presenting a detailed investigation into various methods of incorporating syntactic knowledge into neural SRL models, finding it consistently beneficial.

Document Level Consistency

Document-level modelling has been shown to be beneficial for NLP tasks such as machine summarization (Chen et al. 2016), translation (Maruf and Haffari 2018; Voita et al. 2018; Junczys-Dowmunt 2019), sentiment analysis (Bhatia, Ji, and Eisenstein 2015), and question answering (Verberne et al. 2007; Sadek and Meziane 2016). For semantic analyzers, document-level consistency is an important requirement. Indeed, when training on complete documents, it also provides

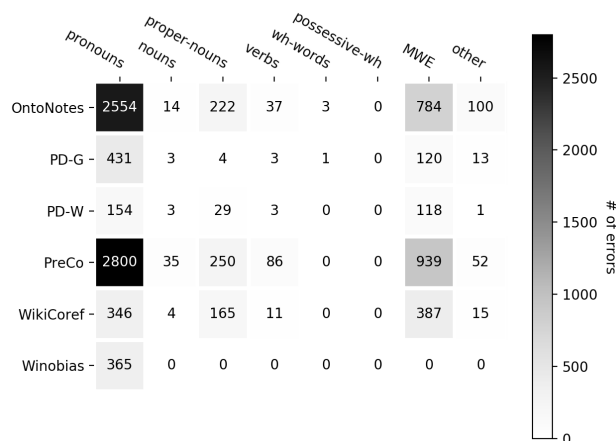


Figure 6: Heatmap showing the POS-tag categories for the antecedents that our fine-tuned coreference system incorrectly classified. All domains except WikiCoref have the highest amount of errors made when the antecedent is a pronoun. Here, pronouns are PRP, PRP\$; MWE is any multi-word expression, nouns are NN, NNS; proper-nouns are NNP, NNPS; verbs are VB, VBD, VBG, VBN, VBP, VBZ; other tags we observed were IN, JJR, JJ, RB, DT, CD, MD, POS; and wh-words are WDT, WRB, WP, WPS\$.

a strong input signal. In previous work Tang, Qin, and Liu (2015) presented a user product neural network and validated the effects of users and products in terms of sentiment and text-based consistency. Likewise, Du et al. (2019) used label consistency as an additional objective for a procedural text comprehension model, showing state-of-the-art performance. More recently, Liu and Lapata (2018) used discourse structure and global consistency to guide a machine comprehension model.

Our approach is orthogonal and possibly complementary to those described above: we investigate the consistency in the overall information presented in complete documents for span graphs composed of semantic role labeling and coreference resolution annotations.

Conclusion

We presented a joint coreference resolver and semantic role labeler along with a method of fine-tuning them with document-level coherence rewards over unlabeled documents. We find that this leads to considerable performance gains for coreference resolution across domains, and moderate improvements for semantic role labeling. Results are presented across six English coreference resolution datasets and four English semantic role labeling datasets. Our code will be made publicly available at <https://github.com/rahular/joint-coref-srl>

Future work will improve the efficiency of our training procedure to allow fine-tuning on longer documents, and investigate how the models can be further improved with better credit assignment.

	OntoNotes		PreCo		PD-G		PD-W		WikiCoref		WinoBias		Average	
	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours
LSTM + CNN	62.58	62.59	42.36	44.37	46.99	46.73	32.72	33.24	39.80	39.82	69.61	69.68	49.01	49.40
BERT-Tiny	61.01	61.35	42.26	45.41	48.44	48.93	37.65	37.76	45.16	45.03	63.68	67.23	49.70	50.95
BERT-Mini	64.20	64.15	44.82	46.07	49.95	50.31	40.22	40.26	49.70	49.44	66.76	67.07	52.61	52.88
BERT-Small	65.81	66.39	44.32	47.51	51.25	52.14	42.35	42.11	50.30	50.55	62.53	64.72	52.76	53.90
BERT-Medium	68.46	68.72	45.85	48.53	54.65	54.68	42.41	41.80	54.05	54.20	68.57	69.23	55.67	56.19
BERT-Base	71.48	71.53	46.89	48.61	56.87	57.27	44.79	44.16	55.02	55.32	71.62	72.17	57.78	58.18

Table 5: COREFERENCE RESOLUTION results of single-task models. ‘Baseline’ and ‘Ours’ indicate the average F_1 scores of MUC, B^3 and CEAF $_{\phi_4}$ for the supervised baseline and coherence fine-tuned models respectively. PD-(G/W) — Phrase Detectives (Gutenberg/Wikipedia) splits.

	OntoNotes		PreCo		PD-G		PD-W		WikiCoref		WinoBias		Average	
	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours
LSTM + CNN	62.13	62.15	42.77	47.66	47.06	47.10	35.23	35.54	40.47	41.01	64.23	64.13	48.65	49.60
BERT Tiny	59.76	60.53	42.22	49.11	42.58	42.22	35.46	35.53	46.68	47.90	47.23	71.73	45.65	51.17
BERT Mini	63.43	63.80	44.18	46.40	47.11	47.56	38.95	39.09	51.32	51.89	55.88	69.45	50.14	53.02
BERT Small	65.40	65.75	44.91	46.82	51.29	51.42	41.33	40.72	51.72	52.24	52.89	65.30	51.26	53.73
BERT Medium	67.70	68.06	45.99	47.52	53.65	53.21	42.65	42.80	52.94	53.30	45.97	68.46	51.48	55.52
BERT Base	70.78	71.23	47.29	48.23	55.46	55.32	43.80	43.50	57.78	57.53	63.29	69.62	56.40	57.55

Table 6: COREFERENCE RESOLUTION results of multi-task models. ‘Baseline’ and ‘Ours’ indicate the average F_1 scores of MUC, B^3 and CEAF $_{\phi_4}$ for the supervised baseline and coherence fine-tuned models respectively. PD-(G/W) — Phrase Detectives (Gutenberg/Wikipedia) splits.

	OntoNotes		Conll05-WSJ		Conll05-Brown		EWT		Average	
	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours
LSTM + CNN	72.14	72.14	67.33	67.72	63.41	63.26	67.64	67.84	67.63	67.74
BERT-Tiny	65.05	65.01	53.02	52.91	51.62	52.56	57.79	57.83	56.87	57.08
BERT-Mini	77.32	77.34	68.74	68.59	65.75	65.46	70.22	70.53	70.51	70.48
BERT-Small	81.21	81.21	73.11	73.45	69.80	70.42	72.91	72.85	74.26	74.48
BERT-Medium	82.30	82.32	75.24	75.21	70.21	69.96	74.73	74.79	75.62	75.57
BERT-Base	85.88	85.97	78.93	78.83	75.01	75.30	78.00	77.99	79.46	79.52

Table 7: SEMANTIC ROLE LABELING results of single-task models. ‘Baseline’ and ‘Ours’ indicate the average token F_1 scores of the supervised baseline and coherence fine-tuned models respectively.

	OntoNotes		Conll05-WSJ		Conll05-Brown		EWT		Average	
	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours
LSTM + CNN	72.92	72.58	68.02	67.82	60.98	60.52	67.21	67.27	67.28	67.05
BERT Tiny	63.91	64.03	52.16	52.19	52.60	52.97	57.94	58.23	56.65	56.85
BERT Mini	77.68	77.69	66.10	66.19	69.83	69.86	70.78	70.77	71.10	71.13
BERT Small	81.08	81.10	69.89	70.07	73.45	73.67	74.48	74.25	74.72	74.77
BERT Medium	84.45	84.47	73.05	73.35	77.52	77.68	76.55	76.55	77.89	78.01
BERT Base	86.41	86.40	76.59	76.47	79.34	79.30	78.67	78.58	80.25	80.19

Table 8: SEMANTIC ROLE LABELING results of multi-task models. ‘Baseline’ and ‘Ours’ indicate the average token F_1 scores of the supervised baseline and coherence fine-tuned models respectively.

References

- Abend, O.; and Rappoport, A. 2013. Universal Conceptual Cognitive Annotation (UCCA). In *ACL*, 228–238. Sofia, Bulgaria. URL <https://aclweb.org/anthology/P13-1023>.
- Abend, O.; and Rappoport, A. 2017. The State of the Art in Semantic Representation. In *ACL*, 77–89. Vancouver, Canada. URL <https://aclweb.org/anthology/P17-1008>.
- Aralikatte, R.; Lent, H.; Gonzalez, A. V.; Herscovich, D.; Qiu, C.; Sandholm, A.; Ringaard, M.; and Sjøgaard, A. 2019. Rewarding Coreference Resolvers for Being Consistent with World Knowledge. In *EMNLP-IJCNLP*, 1229–1235. Hong Kong, China. URL <https://aclweb.org/anthology/D19-1118>.
- Banarescu, L.; Bonial, C.; Cai, S.; Georgescu, M.; Griffitt, K.; Hermjakob, U.; Knight, K.; Koehn, P.; Palmer, M.; and Schneider, N. 2013. Abstract Meaning Representation for Sembanking. In *LAW*, 178–186. Sofia, Bulgaria. URL <https://aclweb.org/anthology/W13-2322>.
- Bhatia, P.; Ji, Y.; and Eisenstein, J. 2015. Better Document-level Sentiment Analysis from RST Discourse Parsing. In *EMNLP*, 2212–2218. Lisbon, Portugal. URL <https://aclweb.org/anthology/D15-1263>.
- Bos, J.; Basile, V.; Evang, K.; Venhuizen, N. J.; and Bjerva, J. 2017. The Groningen meaning bank. In *Handbook of linguistic annotation*, 463–496. Springer.
- Cai, D.; and Lam, W. 2020. AMR Parsing via Graph-Sequence Iterative Inference. In *ACL*, 1290–1301. Online. URL <https://aclweb.org/anthology/2020.acl-main.119>.
- Carreras, X.; and Màrquez, L. 2005. Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *CoNLL*, 152–164. Ann Arbor, Michigan. URL <https://aclweb.org/anthology/W05-0620>.
- Chen, H.; Fan, Z.; Lu, H.; Yuille, A.; and Rong, S. 2018. PreCo: A Large-scale Dataset in Preschool Vocabulary for Coreference Resolution. In *EMNLP*, 172–181. Brussels, Belgium. URL <https://aclweb.org/anthology/D18-1016>.
- Chen, Q.; Zhu, X.; Ling, Z.; Wei, S.; and Jiang, H. 2016. Distraction-based neural networks for document summarization. *arXiv:1610.08462*.
- Choshen, L.; Fox, L.; Aizenbud, Z.; and Abend, O. 2020. On the Weaknesses of Reinforcement Learning for Neural Machine Translation. In *ICLR*. URL <https://openreview.net/forum?id=H1eCw3EKvH>.
- Clark, K.; and Manning, C. D. 2016a. Deep Reinforcement Learning for Mention-Ranking Coreference Models. In *EMNLP*, 2256–2262. Austin, Texas. URL <https://aclweb.org/anthology/D16-1245>.
- Clark, K.; and Manning, C. D. 2016b. Improving Coreference Resolution by Learning Entity-Level Distributed Representations. In *ACL*, 643–653. Berlin, Germany. URL <https://aclweb.org/anthology/P16-1061>.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 4171–4186. Minneapolis, Minnesota. URL <https://aclweb.org/anthology/N19-1423>.
- Du, X.; Dalvi, B.; Tandon, N.; Bosselut, A.; Yih, W.-t.; Clark, P.; and Cardie, C. 2019. Be Consistent! Improving Procedural Text Comprehension using Label Consistency. In *NAACL*, 2347–2356. Minneapolis, Minnesota. URL <https://aclweb.org/anthology/N19-1244>.
- Emami, A.; Trischler, A.; Suleman, K.; and Cheung, J. C. K. 2018. A Generalized Knowledge Hunting Framework for the Winograd Schema Challenge. In *NAACL*, 25–31. New Orleans, Louisiana, USA. URL <https://aclweb.org/anthology/N18-4004>.
- Ghaddar, A.; and Langlais, P. 2016. WikiCoref: An English Coreference-annotated Corpus of Wikipedia Articles. In *LREC*, 136–142. Portorož, Slovenia. URL <https://aclweb.org/anthology/L16-1021>.
- Hacioglu, K. 2004. Semantic role labeling using dependency trees. In *COLING*, 1273–1276.
- Haghighi, A.; and Klein, D. 2009. Simple coreference resolution with rich syntactic and semantic features. In *EMNLP*, 1152–1161. ACL.
- He, L.; Lee, K.; Levy, O.; and Zettlemoyer, L. 2018. Jointly Predicting Predicates and Arguments in Neural Semantic Role Labeling. In *ACL*, 364–369. Melbourne, Australia. URL <https://aclweb.org/anthology/P18-2058>.
- He, L.; Lee, K.; Lewis, M.; and Zettlemoyer, L. 2017. Deep Semantic Role Labeling: What Works and What’s Next. In *ACL*, 473–483. Vancouver, Canada. URL <https://aclweb.org/anthology/P17-1044>.
- Herscovich, D.; Abend, O.; and Rappoport, A. 2017. A Transition-Based Directed Acyclic Graph Parser for UCCA. In *ACL*, 1127–1138. Vancouver, Canada. URL <https://aclweb.org/anthology/P17-1104>.
- Joshi, M.; Levy, O.; Zettlemoyer, L.; and Weld, D. 2019. BERT for Coreference Resolution: Baselines and Analysis. *EMNLP*.
- Junczys-Dowmunt, M. 2019. Microsoft translator at wmt 2019: Towards large-scale document-level neural machine translation. *arXiv:1907.06170*.
- Kamp, H.; and Reyle, U. 2013. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*. Springer.
- Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *ICLR*. URL <http://arxiv.org/abs/1412.6980>.
- Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*. OpenReview.net. URL <https://openreview.net/forum?id=SJU4ayYgl>.
- Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; and Hovy, E. 2017. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In *EMNLP*, 785–794. Copenhagen, Denmark. URL <https://aclweb.org/anthology/D17-1082>.
- Langford, J.; and Zadrozny, B. 2005. Relating Reinforcement Learning Performance to Classification Performance. In *ICML*.

- Lee, K.; He, L.; Lewis, M.; and Zettlemoyer, L. 2017. End-to-end Neural Coreference Resolution. In *EMNLP*, 188–197. Copenhagen, Denmark. URL <https://aclweb.org/anthology/D17-1018>.
- Lee, K.; He, L.; and Zettlemoyer, L. 2018. Higher-Order Coreference Resolution with Coarse-to-Fine Inference. In *NAACL*, 687–692. New Orleans, Louisiana. URL <https://aclweb.org/anthology/N18-2108>.
- Levin, B. 1993. *English verb classes and alternations: A preliminary investigation*. University of Chicago press.
- Liu, J.; Cohen, S. B.; and Lapata, M. 2018. Discourse Representation Structure Parsing. In *ACL*, 429–439. Melbourne, Australia. URL <https://aclweb.org/anthology/P18-1040>.
- Liu, Y.; and Lapata, M. 2018. Learning Structured Text Representations. *TACL* 63–75. URL <https://aclweb.org/anthology/Q18-1005>.
- Maruf, S.; and Haffari, G. 2018. Document Context Neural Machine Translation with Memory Networks. In *ACL*, 1275–1284. Melbourne, Australia. URL <https://aclweb.org/anthology/P18-1118>.
- Ng, V. 2007. Shallow Semantics for Coreference Resolution. In *IJCAI*, volume 2007, 1689–1694.
- O’Gorman, T.; Regan, M.; Griffitt, K.; Hermjakob, U.; Knight, K.; and Palmer, M. 2018. AMR Beyond the Sentence: the Multi-sentence AMR corpus. In *COLING*, 3693–3702. Santa Fe, New Mexico, USA. URL <https://aclweb.org/anthology/C18-1313>.
- Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*, 1532–1543. Doha, Qatar. URL <https://aclweb.org/anthology/D14-1162>.
- Poesio, M.; Chamberlain, J.; Kruschwitz, U.; Robaldo, L.; and Ducceschi, L. 2013. Phrase Detectives: Utilizing Collective Intelligence for Internet-Scale Language Resource Creation. *ACM Trans. Interact. Intell. Syst.* (1).
- Ponzetto, S. P.; and Strube, M. 2006a. Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution. In *NAACL*, 192–199. New York City, USA. URL <https://aclweb.org/anthology/N06-1025>.
- Ponzetto, S. P.; and Strube, M. 2006b. Semantic Role Labeling for Coreference Resolution. In *Demonstrations*. URL <https://aclweb.org/anthology/E06-2015>.
- Pradhan, S.; Hacioglu, K.; Ward, W.; Martin, J. H.; and Jurafsky, D. 2005. Semantic role chunking combining complementary syntactic views. In *CoNLL*, 217–220.
- Pradhan, S.; Moschitti, A.; Xue, N.; Uryupina, O.; and Zhang, Y. 2012. CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes. In *CoNLL*, 1–40. Jeju Island, Korea. URL <https://aclweb.org/anthology/W12-4501>.
- Prange, J.; Schneider, N.; and Abend, O. 2019. Semantically Constrained Multilayer Annotation: The Case of Coreference. In *DMR*, 164–176. Florence, Italy. URL <https://aclweb.org/anthology/W19-3319>.
- Punyakanok, V.; Roth, D.; and Yih, W.-t. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics* (2): 257–287.
- Rahman, A.; and Ng, V. 2011. Coreference resolution with world knowledge. In *ACL*, 814–824. ACL.
- Sadek, J.; and Meziane, F. 2016. A discourse-based approach for Arabic question answering. *TALLIP* (2): 1–18.
- Shibata, T.; and Kurohashi, S. 2018. Entity-Centric Joint Modeling of Japanese Coreference Resolution and Predicate Argument Structure Analysis. In *ACL*, 579–589. Melbourne, Australia. URL <https://aclweb.org/anthology/P18-1054>.
- Silveira, N.; Dozat, T.; de Marneffe, M.-C.; Bowman, S.; Connor, M.; Bauer, J.; and Manning, C. 2014. A Gold Standard Dependency Corpus for English. In *LREC*, 2897–2904. Reykjavik, Iceland. URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/1089_Paper.pdf.
- Strubell, E.; Verga, P.; Andor, D.; Weiss, D.; and McCallum, A. 2018. Linguistically-Informed Self-Attention for Semantic Role Labeling. In *EMNLP*, 5027–5038. Brussels, Belgium. URL <https://www.aclweb.org/anthology/D18-1548>.
- Sutton, C.; and McCallum, A. 2005. Joint parsing and semantic role labeling. Technical report, MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER SCIENCE.
- Tang, D.; Qin, B.; and Liu, T. 2015. Learning Semantic Representations of Users and Products for Document Level Sentiment Classification. In *ACL*, 1014–1023. Beijing, China. URL <https://aclweb.org/anthology/P15-1098/>.
- Turc, I.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Well-Read Students Learn Better: On the Importance of Pre-training Compact Models.
- Veličković, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2018. Deep graph infomax. *arXiv:1809.10341*.
- Verberne, S.; Boves, L.; Oostdijk, N.; and Coppen, P. 2007. Discourse-based answering of why-questions. *TAL*.
- Voita, E.; Serdyukov, P.; Sennrich, R.; and Titov, I. 2018. Context-Aware Neural Machine Translation Learns Anaphora Resolution. In *ACL*, 1264–1274. Melbourne, Australia. URL <https://aclweb.org/anthology/P18-1117>.
- Wiseman, S.; Rush, A. M.; and Shieber, S. 2016. Antecedent Prediction Without a Pipeline. In *CORBON*, 53–58. San Diego, California. URL <https://aclweb.org/anthology/W16-0708>.
- Xia, Q.; Li, Z.; Zhang, M.; Zhang, M.; Fu, G.; Wang, R.; and Si, L. 2019. Syntax-aware neural semantic role labeling. In *AAAI*, 7305–7313.
- Zhang, H.; Song, Y.; and Song, Y. 2019. Incorporating Context and External Knowledge for Pronoun Coreference Resolution. In *NAACL*, 872–881. Minneapolis, Minnesota. URL <https://aclweb.org/anthology/N19-1093>.
- Zhao, J.; Wang, T.; Yatskar, M.; Ordonez, V.; and Chang, K.-W. 2018. Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods. In *NAACL*, 15–20. New Orleans, Louisiana. URL <https://aclweb.org/anthology/N18-2003>.