EFFICIENT NEURAL METHODS FOR COREFERENCE RESOLUTION

by

Patrick Xia

A dissertation submitted to The Johns Hopkins University

in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

October, 2022

Abstract

Coreference resolution is a core task in natural language processing and in creating language technologies. Neural methods and models for automatically resolving references have emerged and developed over the last several years. This progress is largely marked by continuous improvements on a single dataset and metric. In this thesis, the assumptions that underlie these improvements are shown to be unrealistic for real-world use due to the computational and data tradeoffs made to achieve apparently high performance. The thesis outlines and proposes solutions to three issues. First, to address the growing memory requirements and restrictions on input document length, a novel, constant memory neural model for coreference resolution is proposed and shown to attain performance comparable to contemporary models. Second, to address the failure of these models to generalize across datasets, continued training is evaluated and shown to be successful for transferring coreference resolution models between domains and languages. Finally, to combat the gains obtained via the use of increasingly large pretrained language models, multitask model pruning can be applied to maintain a single (small) model for multiple datasets. These methods

reduce the computational cost of running a model and the annotation cost of creating a model for any arbitrary dataset. As real-world applications continue to demand resolution of coreference, methods that reduce the technical cost of training new models and making predictions are greatly desired, which this thesis addresses.

Primary Reader and Advisor: Benjamin Van Durme

Secondary Readers: Kenton Lee & Kenton Murray

Dedication

To my parents, my sisters, and Baltimore.

Acknowledgments

I would like to first thank my advisor, Benjamin Van Durme, for everything. For advising that first project. For his technical and historical knowledge around NLP and linguistics. For the necessary softer research skills: precise communication, perseverance, and paradoxically, when to give up. For continuing to champion and encouraging my wildest ideas and having faith in my pursuits despite their disconnectedness to each other and countless paper rejections. For teaching me about the bigger picture both academically and in my career. While NLP fads will come and go, those softer lessons will stick with me forever.

Next, I'd like to thank several mentors who have shaped the technical direction of this thesis and my PhD: João Sedoc's mentorship led to the first findings in this thesis. As we transitioned to virtual at the start of the COVID-19 pandemic, he consistently continued to encourage new ideas and exploration and kept me well-motivated. Kenton Lee provided expert insights and intuition over the years on this topic and inspiring me to appreciate readable "research code". Additionally, discussions with Kenton Murray, Aaron Steven White, Kyle Rawlins, Michelle Yuan, Shubham Toshniwal, and

ACKNOWLEDGMENTS

Huda Khayrallah significantly steered the trajectory of this work.

I had the fortune of working on and learning from several other projects:

David Yarowsky introduced me, a new researcher, to paper writing and the academic community. He encouraged me to explore multiple areas and reminded me to look beyond one project. In addition, his emphasis on multilinguality and the applicability of non-neural methods has stuck with me. Through David, I also had the pleasure to work with Chris Kirov, John Sylak-Glassman, and Ryan Cotterall, all of whom shaped my early research experience.

Working on a large (10+ people) team for the 2018 JSALT workshop studying the trendiest topic in NLP is a rare experience I am extremely lucky to have had. Our team leads, Ellie Pavlick and Sam Bowman, demonstrated an approach to research and writing (specifically, how to pivot and the unformulaicness of paper writing) that now guides my research. In addition, I enjoyed the rapid pace and close collaborations with Ian Tenney and Alex Wang and the speculative discussions with the rest of the team: Najoung Kim, Roma Patel, Tom McCoy, Jan Hula, Alexis Ross, Berlin Chen, Shuning Jin, Raghu Pappagari, Yinghui Huang, Katherin Yu, Edouard Grave, Dipanjan Das, and Tal Linzen.

My internship mentors, William Cohen and Ming-Wei Chang at Google and Richard Shin at Microsoft, showed me how difficult it is to convert between research and industry problems. Through them, I learned what "big" means in NLP and affirmed my affinity towards real-world problems.

ACKNOWLEDGMENTS

The environment at the Center for Language and Speech Processing (CLSP) at Johns Hopkins played a central role in my PhD. In addition to the aforementioned researchers, I had the chance to work with and learn from many additional researchers at CLSP, all of whom motivated me in my research journey: Nicholas Andrews, Anton Belyy, Tongfei Chen, Yunmo Chen, Ryan Culkin, Shuoyang Ding, Mark Dredze, Seth Ebner, Jason Eisner, Yukun Feng, Craig Harman, Edward Hu, Huda Khayrallah, Xutai Ma, Marc Marone, Chandler May, Arya McCarthy, Sabrina Mielke, Adam Poliak, Matt Post, Kyle Rawlins, Guanghui Qin, Suzanna Sia, Abhinav Singh, Nathaniel Weir, Shinji Watanabe, Shijie Wu, Mahsa Yarmohammadi, Boyuan Zheng.

Of course, CLSP would not be possible without our wonderful administrative and IT staff over the years: Ruth Scally, Jennifer Linton, Lauren MacNeil, Carl Pupa, Kim Franklin, and Zack Burwell. Ruth knew the answers to every questions and was always uplifting to talk to; she was a pillar I leaned on for the entirety of my PhD.

CLSP was also my second home. Huda, Rebecca, Adam, Zach, Seth, Nathaniel, Rachel, Elliot, Marc, and Neha livened up the office and created a vibrant and warm community that I would look forward to and will miss. My first-year cohort (Zach, Matthew, Pamela, Teodor, Yasamin, Becky, Adarsh, Hongyuan) helped me find a community in a new city. Others at JHU contributed to a healthy balance between work and life: Winston and Hainan for musical and nostalgic grocery excursions, Seth, Eric, Walter, Michael, and Audrey for my halfhearted return to quiz bowl, and Elliot, Pamela, Mitchell, Marc, Neha, Drew, Lawrence, and Tim for climbing company.

ACKNOWLEDGMENTS

When I was an undergrad at CMU, Chris Dyer, David Bamman, and Noah Smith kicked me off on this journey for teaching a phenomenal NLP class and supervising my independent research projects. I know I was far from their best undergrad researcher, yet without those experiences and their mentorship, I would not have followed the path I did. I will be forever grateful for their teaching and advising.

The ever-onlineness of my friends made living in Baltimore feel less distant (from them), especially during my first years. In particular, I appreciate the camaraderie from Chris Jones and Will Crichton throughout our journeys, from when we first decided to apply to grad school to when we wrote our dissertations. In addition, Galactic Puzzle Hunt and teammate scratched my itch to write and solve puzzles[1] and was an online community that was especially valuable when we retreated back to our homes during the pandemic. I thank all my friends over the years[♡] for the lively virtual work sessions, games, and conversations late into the night.

Finally, this would not have been possible without the love from my family. My sisters (Mary and Rosa) and my parents (Hua Xiang and Jihong) never stopped supporting me in my endeavors down in Baltimore, notwithstanding the countless road trips they made there to visit me.

[1] https://blog.vero.site/post/puzzlehunts.
[♡] {GW, JJ, TY, JH, XZ, JW, AT, MB, AW, LC, CW, WSL, BL, IW, JS, MS, BS, AI, ...}

Contents

Abstract **ii**

Acknowledgments **v**

List of Tables **xv**

List of Figures **xix**

1 Introduction **1**

1.1 Motivation 2

1.2 Contributions and Organization 4

2 Background: Coreference resolution **5**

2.1 Coreference Resolution 6

2.2 Datasets and Annotation 10

2.2.1 Datasets 10

2.2.2 Annotation 16

2.3 Evaluation of coreference systems 18

2.3.1 Quantitative Evaluation 18

2.3.2 Qualitative Analysis 23

2.4 Automatic systems 23

2.4.1 Sieve-based approach 24

2.4.2 Learned features 25

2.5 End-to-end neural models 27

2.5.1 End-to-end (E2E) model 28

2.5.2 Variations of neural end-to-end models 32

2.6 Remarks 39

3 Motivation: Coreference resolution in practice 46

3.1 Recovering Singletons 47

3.1.1 Abstract 48

3.1.2 Background 48

3.1.3 Augmenting Mention Annotations 53

3.1.4 Experiments and Results 55

3.1.5 Discussion and weaknesses 58

3.2 A coreference pipeline in information extraction in practice 60

3.2.1 Streaming Multimedia Knowledge Base Population (SM-KBP) 60

3.2.2 Discussion 64

3.3 Takeaways 65

4 Efficient inference of coreference resolution models **68**

4.1 Constant memory coreference resolution 69

4.1.1 Background 70

4.1.2 Model 71

4.1.3 Experiments 75

4.1.4 Results 77

4.1.4.1 Performance 77

4.1.4.2 Document Length 77

4.1.5 Inference Memory 78

4.1.6 Segment Length 81

4.1.7 Span Representations 81

4.1.8 Technical Appendix 83

4.1.9 Conclusions 85

4.2 Adapting ICoref beyond OntoNotes 86

4.3 Low-latency online coreference resolution 90

4.3.1 Motivation and Introduction 91

4.3.2 Task: Online Coreference Resolution 93

4.3.3 Method 95

4.3.3.1 Datasets 95

4.3.3.2 Models 97

4.3.4 Experiments and Results 98

4.3.4.1 Masking the future 99
4.3.4.2 Online inference strategies 99
4.3.4.3 Error correction with rollback 103
4.3.4.4 Latency analysis 103
4.3.5 Technical Appendix 105
4.3.6 Conclusions and Limitations 106

5 Improving data efficiency via model transfer 108

5.1 A multilingual information extraction system 109
5.1.1 An overview of Large Ontology Multilingual Extraction (LOME) 110
5.1.2 Multilingual Coreference Resolution 113
5.2 Model transfer 117
5.2.1 Introduction 118
5.2.2 Coreference Resolution 120
5.2.3 Methods 121
5.2.3.1 Continued Training 121
5.2.3.2 Incremental Coreference Model 122
5.2.3.3 Data 123
5.2.3.4 Source models 126
5.2.4 Experiments and Results 128
5.2.4.1 How effective is continued training for domain adaptation? 129

5.2.4.2 How to allocate annotated documents? 134

5.2.4.3 How much do the source models forget? 136

5.2.4.4 Which encoder layers are important? 137

5.2.5 Conclusion and Limitations 139

6 Reducing model size **141**

6.1 A brief overview of model compression 142

6.2 Multitask Model Pruning 145

6.2.1 Introduction and Background 146

6.2.2 Approach 147

6.2.2.1 Pruning Methods 148

6.2.2.2 Multitask extension 151

6.2.2.3 Model and Datasets 152

6.2.3 Experiments and Results 153

6.2.3.1 Comparison of pruning methods 153

6.2.4 Multitask pruning 156

6.2.5 Auxiliary multitask pruning objective 157

6.2.6 Discussion and Limitations 158

6.3 Reducing model size for multiple coreference datasets 159

6.3.1 Background 160

6.3.2 Methods 161

6.3.3 Proof of concept 162

6.3.4 Discussion 164

6.4 Future directions 166

7 Conclusions and Future Directions **169**

7.1 Summary 170

7.2 Impact 170

7.3 Short-term future directions 172

7.4 Rethinking the "task" of coreference 177

7.5 Closing Remarks 181

Vita **248**

List of Tables

2.1 Number of documents for each of the datasets. A "document" varies substantially; in some cases, they are only sentences while in others, each document can contain thousands of tokens. This is not an exhaustive list, but contains several commonly-used datasets, including those used in this work. 12

2.2 These examples from different datasets illustrate the differences in annotation standards, specifically for what is markable as a mention. Mentions are bracketed and entity clusters are subscripted with the same number. Table reproduced from Xia and Van Durme (2021). . . 13

2.3 This table lists a couple of examples of non-English (collections of) datasets, along with their dataset statistics. The table is far from exhaustive. The statistics for ANCOR^{fr} and RuCor^{ru} may not be accurate as what is available and what is reported across papers sometimes differ. ANCOR^{fr} reports 500K tokens in total. 14

2.4 Some examples of probing datasets. Most of the examples in these datasets are sentences or sentence pairs. For edge probing, each example targets a different mention, and so each sentence often occurs multiples in the dataset. 15

2.5 This table, reproduced from Raghunathan et al. (2010), shows what rules are used for each pass of the sieve and the average (MUC and B^3) precision, recall, and F1 on the ACE2004-ROTH-DEV set (Bengtson and Roth, 2008). "Type" refers to the types of antecendents (**P**ronominal vs. **N**ominal) that are targeted by that pass. The sieve targets initially high-precision rules and the later passes trade off precision for recall, ultimately boosting F1. In pass 2, two of the constraints need to be satisfied; these constraints are determined by syntactic parsers, acronym detection algorithm, an Wikipedia demonyms. not-i-within-i refers to one NP not being a child of another NP in the antecedent cluster. Pronouns are enforced by number, gender, person, animacy, and entity type agreement. 26

2.6 Claims of state of the art results on OntoNotes 5.0 (which is the CoNLL-2012 shared task), starting around 2015 with a few pipeline models followed by the E2E models and several of its variations. LF = learned features; e2e = end-to-end; hoi = higher order inference; PLM = pretrained language models or encoders; QA = question answering reformulation; word = word-based models; syntax = incorporating gold syntactic annotations at training. 41

3.1 Precision and recall of several r-span configurations treating OntoNotes coreference cluster spans as ground truth. 54

3.2 Precision and recall for r-span and gold mention detection. MD is the span-based single-task mention detector. Gold numbers show that the current models all overgenerate candidate spans; the numbers are also comparable to Table 3.1. 56

3.3 Performance in a pipeline and multitask setting on the OntoNotes 5.0 test set. Pipelined models (+) are given spans as input. l spans are from Lee et al. (2018); r spans are from Multitask; r_g are the gold r-spans derived from gold annotations; and g are the gold coreference spans. A linking-only model does not train a mention scorer for the first beam step. Nothing is bolded because most rows are not comparable as some models use gold spans. 58

4.1 Complete results of our model on the OntoNotes 5.0 test set with three coreference resolution metrics: MUC, B^3, and $CEAF_{\phi_4}$. For completeness, scores for the contemporaneous (and still current) state-of-the-art are included. All models use an encoder derived from SpanBERT-large. 77

4.2 Average F1 score on the development set broken down by document length and number of speakers. **JS-L** refers to the `spanbert_large` model from Joshi et al. (2020), which is the baseline, and -evict refers to the model without eviction. 78

4.3 Space needed and performance over the development set. **JS-B** and **JS-L** refer to the `base` and `large` variants SpanBERT used in the baseline. 79

4.4 Average dev. F1 score for models trained and evaluated across a range of segment lengths (either fixed number of sentences or subtokens). . 81

4.5 A model is trained with and without sentence-level causal attention masks. This table reports the difference in F1 between inference with and without these mask in the offline setting. The numerical results are also reported in Table 4.6. 100

4.6 This is the full version of Table 4.5, on the test set. Each entry instead shows the score with mask and the score without mask instead of the difference. 101
4.7 Final F1, running F1, and wait time for each datasets and four inference algorithms. The proposed rollback mechanism offers a strong compromise with higher F1s and comparable wait times vs. the fastest online models, and a final F1 comparable to offline ICoref. Naive online C2F is the strongest method, but also the slowest. 102
4.8 The edits made in each dev set via rollback are categorized: **Ment**ion detection errors, missed **New** clusters, and incorrect links to **Existing** clusters. We report the percentage of (wrong→right, right→wrong) edits. The unreported fraction of edits are wrong→wrong. 103

5.1 Average F1 scores by language trained with gold mentions, with and without Russian data in training. The superscripts o indicates data from OntoNotes 5.0 (dev), s indicates data from SemEval 2010 Task 1 (dev), and a is the AnCor data (test). Note: this table differs from that of Xia et al. (2021), as that paper reports numbers specifically for the model checkpoint in the demo. For reasons unclear to me now, the model included in the demo performs slightly worse in most languages. 115
5.2 Average F1 scores by language trained without gold mentions, with and without a finetuned *XLM-R* encoder. The superscripts o indicates data from OntoNotes 5.0 (dev), s indicates data from SemEval 2010 Task 1 (dev). 117
5.3 Number of documents for each of the datasets considered in this work. For the smaller datasets, we perform k-fold cross-validation. 123
5.4 Statistics of markables that are either reduced or ignored from the preprocessing of ARRAU^{RST} to convert it into a format consistent with the ICoref model used for the other datasets in this work. 126
5.5 Training set sizes considered for each dataset. *For SARA, the entire fold is used, which contains 138 documents on average. 127
5.6 Test F_1 on all datasets and the previous state-of-the-art on each dataset, to the best of our knowledge. Again, the goal is to benchmark the general method of continued training described in this study, which will not necessarily outperform models that incorporate domain or language specific knowledge. The best Transfer model is determined by the dev set. *ARRAU^{RST} is not directly comparable to prior work as it is test on a slightly differently-preprocessed subset. Multi-pass sieve (Raghunathan et al., 2010), Berkeley (Durrett and Klein, 2013), and C2F (Lee et al., 2018) refer to widely-used coreference resolution models. 133

5.7 Dev. F1 scores on each of the models and datasets presented in Table 5.6. For the English dataset, the test score of the model with the best performing score is reported in Table 5.6. 133

6.1 The main variable in our pruning comparison experiments is structure, although we also explore two other variables. There are 3 variables with 2 choices each, leading to $2^3 = 8$ combinations for model settings. 148

6.2 Individual task performance (dev.) of a model pruned using rank pruning with the multitask objective. The size of the training set for each task is also listed. RP (9) is trained on all nine tasks, RP (3) is the three-task model from Section 6.2.4, and RP (1) represents 9 separate single-task baseline models. BNG (Ben Noach and Goldberg, 2020) is three separate single-task low-rank models tuned using knowledge distillation. 157

6.3 Test F1 of several models on each dataset based on whether they are pruned and which dataset(s) they are trained on. All uses all 4 datasets with separate coreference resolution model parameters for each dataset. All - PreCo omits the (larger) PreCo dataset. Shared is a single model shared across all tasks (and Shared - PreCo omits PreCo). PreCo, QBCoref, LitBank, and Sara are single-task models. Pruning any model further resulted in divergence. 163

List of Figures

2.1 These figures, reproduced from Lee et al. (2017), show the overall framework of end-to-end neural coreference systems. First (top), text is encoded into span representations, in this case, LSTMs are used. Next (bottom), span representations are scored pairwise and against a dummy ϵ "span" which receives a fixed score of 0. For each span representation, the best-scoring preceding span is its label. 29

2.2 In this dialogue from Caroll's *Alice's Adventures in Wonderland*, the pairwise score of (**the soldiers**, **you**) is high, as that is who the Queen is speaking to. In addition, because Alice responds, the score of (**you**, **Alice**) is also high. However, (**the soldiers**, **Alice**) would get a low score. Higher-order inference aims to resolve this puzzle. 33

3.1 (1) contains two entity mentions, either of which could reasonably be referred to later in the discourse: (2A,B) are both plausible continuations. In OntoNotes, (2A) is the true subsequent sentence, and thus the **orange** mentions are both annotated, while the **violet** mention in (1) is a *singleton* referrent, and therefore unannotated in the dataset. 49

3.2 This figure, reproduced from Chen et al. (2019), shows the overall system for our team's second submission to the SM-KBP task. Note that coreference resolution (including Entity Mention Detection and Coreference Resolution) plays a tiny role and is neither the first nor last model. Due to its intermediary role, the coreference model may be optimised more for recall or precision. In this case, higher recall of detected entity mentions is desired, if possible, because subsequent systems can filter unused mentions. 62

4.1 Total size of GPU-allocated tensors for each document in the development set. The base (**JS-B**) and large (**JS-L**) models of the baseline use apparently linear space, while ours with inference segment lengths of 128 and 512 use constant space. 79
4.2 t-SNE plot (left) of span representations of a single document (right) in the development set (cnn_0040_0). Each color/shape is a predicted cluster, while light gray circles indicate predicted singletons. For each span, the gold cluster label (-1, if not annotated) and its contribution to the entity embedding is noted in parentheses. 82
4.3 In this scene from *Friends*, viewers can deduce who "you" refers to at $t = 6$, and we want coreference models to be similarly capable. At $t = 7$, viewers may need more context, such as the identity of the next speaker, to be certain of who "you" refers to. Absent that context for a text-based model, its predictions will be incorrect. *Rollback* is a cheap and local revision mechanism that corrects these type of mistakes. . . 92
4.4 These plots show the average wait time against the final F1 (test) and the running F1 (×) for select models. Varying the update frequency **interpolates** between **online** and **offline** ICoref models in both final F1 and wait time. The **naive online C2F** baseline is also included for comparison. The proposed method of **rollback** offers a strong compromise with higher F1s and comparable wait times vs. the fastest online models, and a final F1 comparable to the offline models. . . . 100
4.5 Simulated mean sentence-level latency given different token arrival rates. 105

5.1 Architecture of LOME, reproduced from Xia et al. (2021). The system processes text documents as input and first uses a FrameNet parser to detect entities and events. Then, a suite of models enrich the entities and events with additional predictions. Each individual model can be trained and tuned independently, ensuring modularity of the pipeline. Annotations between models are transferred using Concrete, a data schema for NLP. 111
5.2 Performance of multilingual coreference linking model as the number of trainable layers in *XLM-R* is increased. 116

5.3 Each subplot shows the test performance for each model and (English) dataset when trained with a different number of documents. The first and second rows are coreference and mention boundary F_1 in the end-to-end setting, while the third row is the coreference F_1 with gold mentions. SpanBERT is a pretrained encoder, while the SpanBERT-On encoders are further finetuned on OntoNotes by Joshi et al. (2020), with **b**ase and **L**arge designating its size. Unlike these (dashed lines) models for which we initialize the encoder, the Transfer models (solid lines) use continued training and initialize the full model with one that has already been trained on a source dataset, either OntoNotes (on) or PreCo (pc). 130
5.4 Like Figure 5.3, this plot demonstates the effectiveness of continued training across different *languages*. XLM-R uses a pretrained encoder (dashed line), while Transfer (en) is first trained on OntoNotes^{en} (solid line). 135
5.5 The expected test F_1 (and standard deviation) on the PreCo dataset for a given number of training documents and 20 sampled subsets of dev documents for two models described in Section 5.2.3.4. The number of runs matching the best full dev checkpoint is in the lower-right. We find that the dev set size has relatively little impact. 136
5.6 Average F_1 of the models on both the target and the original datasets as different number of (target) training examples are used in continued training. The dashed lines are the scores on the target dataset (mirroring Figure 5.3) while the solid lines show performance on the original dataset. 138
5.7 Average F_1 across different models and number of trainable layers, varying between 0, 6, 12 or 24 layers. *Low* vs. *Medium* vs. *All* describes the number of documents used for the first fold of LitBank (10, 40, 80 documents), QBCoref (15, 60, 240 documents), and OntoNotes^{zh} (50, 500, 1810 documents). The initialization methods follow those used throughout the paper. 139

6.1 Left: The performance on MNLI (dev) across [magnitude, movement] and [global, local] pruning strategies. Within each plot, we show the performance of pruned element-wise and rank-pruned models. Right: Comparison of the runtime of a model using rank pruning relative to an entirely dense model, showing that the structured sparsity ensured by rank pruning can lead to practical benefits. *Density* is 1 - sparsity. 153
6.2 Given a fixed parameter budget (expressed as a fraction of a single model size), a single model pruned with a multitask objective (blue) is compared against the best combination of 3 individual task models for a given size. The red line ("mixture") is the Pareto frontier of these combinations. 156

Chapter 1

Introduction

In natural language processing (NLP), one of the core problems is accurately identifying references of pronouns and definite phrases. Consider the following sentence:

Chris visited a doctor and **they** decided to try a new medication.

As readers, we can determine that "**they**" refers to either both "Chris" and the doctor or just "Chris", although it is not inconceivable for "**they**" to instead refer to "a doctor". For decades, linguists and NLP researchers have tackled the challenge of automatically resolving coreference. To date, approaches have become increasingly driven by neural models and large amounts of annotated data.

These advances have led to significant improvements over the methods of even just ten years ago. At the same time, so has the cost of collecting data, training a model, and using the model. This thesis discusses efficient methods to address some of the drawbacks of neural models for coreference resolution.

1.1 Motivation

Coreference resolution is a classic topic within NLP, warranting its own workshop[1] and textbook chapter (Jurafsky and Martin, 2021). Furthermore, resolving coreference has always been valuable within real-world systems. It has been a core component of information extraction tasks (e.g. Grishman and Sundheim (1996)) and in discourse

[1]Fifth Workshop on Computational Models of Reference, Anaphora and Coreference: https://sites.google.com/view/crac2022/

understanding (Elsner and Charniak, 2008). More recently, it has been commercialized[2] and as discussed in Section 3.2, is still an focus in applied works.

In recent years, research has marched on, boasting improvements upon improvements on raw accuracy on one standard evaluation set. While coreference resolution is a classic NLP task grounded in linguistic theory, the strength of a model for it is more aptly measured by how useful it is in specific domains, languages, and when integrated into downstream systems. Therefore, whether or not these models are getting closer to linguistic truth, they are not necessarily climbing the right hill in general.

In fact, the promise of more accurate models for this classic NLP task *does* have several major limitations. For one, the models grow increasingly in size and required compute to run, with the best models requiring 128GB of memory on specialized TPU hardware (Jouppi et al., 2017) to train and 16GB for inference (Wu et al., 2020). Furthermore, these models leverage datasets with thousands of annotations and are restricted to a narrow set of domains and languages. This means that performance in other domains or languages has not kept up, primarily due to the cost of obtaining supervision but also due to specific modeling limitations and decisions.

Therefore, while we are sprinting forward in model accuracy and apparently closing in on human agreement rates, we are leaving everything else behind. This thesis revisits what was left behind – how much did we climb (or descend) on the other hills?

[2]HuggingFace's original chatbot product invested in a neural coref model (https://huggingface.co/coref/).

1.2 Contributions and Organization

This thesis is organized with this introduction (Chapter 1), followed by a comprehensive background on the state of the field of coreference resolution, along with factors that have affected the development of models (Chapter 2). Chapter 3 describes real-world applications that highlight the limitations of many models in the field, motivating a need for solutions that address these issues.

The following chapters cover materials that were previously published:

1. An efficient, constant-memory model (Section 4.1, EMNLP 2020)
2. An online model for inference (Section 4.3, in CRAC 2022)
3. A short study on multilingual coreference resolution (Section 5.1, EACL Demos 2021)
4. A comprehensive study on the success of continued training for domain and language transfer (Section 5.2, EMNLP 2021)
5. A method for reducing model size when there are multiple target tasks (Section 6.2, ENLSP 2021).

Section 4.2 formalizes the model difference between Section 4.1 and the remainder of the thesis, while Section 6.3 presents a proof of concept for pruning a model aimed at multiple coreference resolution datasets. Finally, the thesis concludes (Chapter 7) with several forward-looking future directions and goals for coreference resolution.

Chapter 2

Background: Coreference resolution

This chapter begins with a description of the coreference resolution task (Section 2.1), followed by a discussion on datasets (Section 2.2), evaluation (Section 2.3), and models (Section 2.4, Section 2.5).

2.1 Coreference Resolution

Reference and coreference

Reference is the relationship between expressions; for the purpose of this thesis, we are interested in text. We are interested in *co-reference*, which describes the relationship between two textual expressions that have the same *referent*, which is often either an entity or an action.[1] In each of the following examples, selected spans of text that have the same referent are marked in the same color and bracketed, with the referent written in the subscript.

(1) [**The cat**]$_{\text{CAT}}$ played with [**its**]$_{\text{CAT}}$ toy.

(2) [**Matt and Emma**]$_{\{\text{MATT, EMMA}\}}$ traditionally split a strawberry cake on [**their**]$_{\{\text{MATT, EMMA}\}}$ shared birthday. [**The twins**]$_{\{\text{MATT, EMMA}\}}$ turn 8 this year.

(3) While [**she**]$_{\text{SUMIRE}}$ was traveling with [**Miu**]$_{\text{MIU}}$ on a Greek island,

[1] Coreference in NLP is typically conflated and is often a superset of *anaphora*, where the other referent precedes the current expression (see example (1)). This is contrasted with *cataphora*, where the other referent postcedes the expression (see example (3)). Modern algorithms for coreference cover both phenomena, but both differ with *exophoric* references, where the referenced object is not in the text (e.g. common ground or in other modalities, like visual dialogue (Kottur et al., 2018)).

> [**Sumire**]$_{\text{SUMIRE}}$, [**an aspiring writer**]$_{\text{SUMIRE}}$, suddenly [**disappeared**]$_{\text{DISAPPEARANCE}}$. [**Miu**]$_{\text{MIU}}$ immediately alerted [**her**]$_{\text{SUMIRE?}}$ friend of [[**her**]$_{\text{SUMIRE}}$ **disappearance**]$_{\text{DISAPPEARANCE}}$.

These examples demonstrate some of the common types of coreference. (1) demonstrates pronoun coreference, where "**its**" is coreferent with "**The cat**". In (2), observe that coreference is often a cross-sentence phenomenon, as "**The twins**" and "**Matt and Emma**" are coreferent. In fact, the distance between the two spans can stretch across paragraphs in an article, chapters of a book, or documents in a corpus. Second, coreference can leverage common sense and world knowledge; in (2), it is useful to know that sharing a birthday suggests that Matt and Emma are twins. Along with the lack of other references, this can be used to infer who "**The twins**" refers to. However, coreference can be ambiguous. In (3), it is ambiguous whose friend **Miu** alerted (in "**her** friend"): was it **Miu**'s friend or **Sumire**'s friend?[2] Example (3) also demonstrates a few other common types of coreference: verbs or events (DISAPPEARANCE), appositives or copular structures ("**an aspiring writer**"), and exact string match ("**Miu**").[3]

[2]This example describes the plot from Murakami's *Sputnik Sweetheart*. The ground truth is that he was a mutual friend, so either interpretation would be correct.

[3]Amusingly, even exact string match may not be reliable. Consider the excerpt: "[**Kenton**]$_{\text{KENTON LEE}}$ researched coreference resolution. . . . [**Kenton**]$_{\text{KENTON MURRAY}}$ researched model compression." Resolving this requires real-world knowledge in disambiguating between the two researchers.

Formalization of Coreference Resolution

Here, we define some commonly used terms within the literature for coreference resolution. In the previous examples, each of the bracketed spans of text is a span of interest and is usually referred to as a **mention** (or *markable*) span. Mentions are typically (noun) phrases and pronomials; additional examples that were not highlighted would include "its toy", "a strawberry cake", and "a Greek island." The set of mentions referring to the same referent is a coreference **cluster** or *chain*, like {"The cat", "its"} or {"Miu", "Miu"}. A cluster with a single mention, like {"a strawberry cake"} is called a **singleton**. An **antecedent** of a mention is a coreferring mention that occurs earlier in the text relative to that mention.

Coreference resolution is the task of identifying which spans of text are coreferring. Often, but not always, this simultaneously includes identifying (the boundaries of) the mentions. The task is simple to describe. Essentially, entity (or event) coreference resolution is looking for the answers to:

Q1. (Full coreference) If mentions are not provided: *Find all mentions of each entity (event).*

Q2. (Coreference linking) If mentions are provided: *Do two given spans of text refer to the same entity (event)?*

Formally, the task can be described as: given an input document[4] $D = x_1, \ldots, x_n$,

[4]Cross-document or multi-document coreference is remarked on in Section 2.6.

output the coreference clusters $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ such that each cluster $C_i = \{m_{i,1}, m_{i,2}, ..., m_{i,k_i}\}$ consists of k_i text mentions. Each mention m_j is a set of n_j tokens $m_j = \{x_a, x_{a+1}, ..., x_{a+n_j}\}$.[5] If the mention boundaries are provided (Q2), the task is simpler, as we are instead given both D and a set of mentions, $\mathcal{M}$, and the task is to partition $\mathcal{M} = \bigcup_{C_i \in \mathcal{C}} C_i$ into $\mathcal{C}$.[6]

As illustrated by the examples, some mentions are easy to resolve (via simple heuristics like exact match), while others are difficult or impossible due to missing world knowledge or ambiguities. Automatic systems for coreference resolution face the same challenges for the difficult cases. Further exasperating this issue is the inconsistencies present across annotation guidelines, and therefore datasets. For example, some datasets may be interested in only a subset of entity types, like coarse-grained entity types (people, locations, etc) in literature (Bamman et al., 2020) or specific scientific concepts (drugs, protein names, etc) in biomedical papers (Lu and Poesio, 2021). Other datasets aiming for completeness might further distinguish between the part of speech (event vs. entity), types of links (identical or appositive reference (Hovy et al., 2006; Pradhan et al., 2007)), non-referring mentions (generics or predicative noun phrases (Poesio et al., 2018; Chen et al., 2018)), and traces (implicit or in pro-drop languages (Hovy et al., 2006; Recasens et al., 2010)), or even include bridging (indirect

[5]Typically, each mention consists of contiguous tokens, but this is not always true in the case of *split antecedents*, which is occasionally studied (Yu et al., 2021). Another setting aims to predict *head* words of the mentions rather than the exact boundaries.

[6]There are other possible task descriptions. For example, high-recall *predicted* mention boundaries, like those produced by a syntactic parser, could be provided. One version of this is explored in Section 3.2.

references like "resident" to "building" (Poesio et al., 2018)).

Despite the fine-grained classification of reference types in discourse, the *methods* for automatically resolving coreference share similarities. Even though datasets often blur the boundary between coreference and adjacent relations, these datasets are often bucketed under *coreference resolution*. For the remainder of this work, we simplify the definition of *coreferring expressions* and *mentions* to whatever each dataset uses.

2.2 Datasets and Annotation

Despite the disagreements on what expressions are markable or coreferring, there have been numerous attempts to create datasets for coreference resolution to encourage both data-driven approaches and analysis. These datasets are created by formalizing assumptions into annotation guidelines, which leads to agreement within a single dataset but disagreement across datasets.

2.2.1 Datasets

We first give an overview of the sizes and types of datasets. Tables 2.1, 2.3, and 2.4 list datasets from various years, languages, domains, and sizes. While these lists are not exhaustive, they provide an overview of some of the more commonly used datasets in the field.

Broadly, we can group coreference datasets into three (or four) categories based on

their primary focus. One category is English, document-level coreference resolution. Examples are provided in Table 2.1. Work on these datasets is primarily motivated by improved accuracy at the document level, on the end-to-end task. To do so, innovations typically arise from new modeling contributions, while dataset contributions typically arise from novelty in domain or in size. Furthermore, each dataset creator will make their own domain-dependent decision regarding what spans of text are markable mentions and what is considered a coreference link. Table 2.2 highlights some of the different annotation decisions made for various English datasets. To date, the most studied on dataset is OntoNotes 5.0, as it was used as part of the CoNLL 2012 Shared Task (Pradhan et al., 2012; Weischedel et al., 2013).

Another category of datasets focus on coreference resolution in other languages, some are listed in Table 2.3.[7] Here, datasets are generally smaller than English ones and modeling approaches focus on adapting English models and on language-specific phenomena, such zero anaphora in Chinese (Chen and Ng, 2013), Japanese (Konno et al., 2021), or Italian (Iida and Poesio, 2011). While many models tend to extend from models developed mainly in English (Shibata and Kurohashi, 2018; van Cranenburgh, 2019), there are additional tricks needed to accommodate specific languages. In addition, it is possible for insights gained from multilingual or non-English coreference modeling to transfer to improvements in other languages (like English). There is increased interest in that direction, e.g. the 2022 CRAC multilingual coreference

[7]Recently, Žabokrtský et al. (2022) release a shared task with additional languages and datasets.

Dataset	References	Domain	Training	Dev	Test
OntoNotes^{en}	Pradhan et al. (2012)	Mixed written/spoken texts	2,802	343	348
PreCo	Chen et al. (2018)	Reading comprehension passages	36,120	500	500
LitBank	Bamman et al. (2020)	Literature	80	10	10
QBCoref	Guha et al. (2015)	Quiz questions	240	80	80
ARRAU	Poesio and Artstein (2008), Uryupina et al. (2016), and Uryupina et al. (2020)	Mixed written news	335	18	60
SARA	Holzenberger and Van Durme (2021)	Legal	138	28	28
WikiCoref	Ghaddar and Langlais (2016)	Wikipedia	0	0	30
CI	Chen and Choi (2016), Choi and Chen (2018), and Zhou and Choi (2018)	TV transcripts	987	122	192
OntoGUM	Zhu et al. (2021)	Mixed genres	0	0	168

Table 2.1: Number of documents for each of the datasets. A "document" varies substantially; in some cases, they are only sentences while in others, each document can contain thousands of tokens. This is not an exhaustive list, but contains several commonly-used datasets, including those used in this work.

Dataset	Example	Comments
OntoNotes (general)	Judging from the Americana in [[Haruki Murakami's]$_1$ "A Wild Sheep Chase" [Kodansha]$_2$, 320 pages, $18.95]$_3$, baby boomers on both sides of the Pacific have a lot in common.	*Only coreferring mentions are marked (no singletons).*
ARRAU (news)	Judging from [the Americana in [[Haruki Murakami's]$_1$ "A Wild Sheep Chase" [[Kodansha]$_2$, [320 pages]$_3$, [$18.95]$_4$]$_5$]$_6$]$_7$, [baby boomers on [both sides of [the Pacific]$_8$]$_9$]$_{10}$ have [a lot in [common]$_{11}$]$_{12}$.	*All mentions are marked, even if they are singletons.*
PreCo (general)	[Writer]$_1$: [Ralph Ellison]$_1$ [Novel]$_2$: [Invisible Man]$_2$ [Invisible Man]$_2$ is [[Ellison's]$_1$ best known work]$_2$, most likely because [it]$_2$ was [the only novel [he]$_1$ ever published during [[his]$_1$ lifetime]$_3$]$_2$ and because [it]$_2$ won [him]$_1$ [the National Book Award]$_4$ in [1953]$_5$.	*Singleton mentions are marked. Many documents contain the title as its own sentence.*
LitBank (books)	And [Jo]$_1$ shook the blue army sock till the needles rattled like castanets, and [her]$_1$ ball bounded across [the room]$_2$.	*Only certain ACE categories are marked.*
QBCoref (trivia)	[This author]$_1$ wrote [a play]$_2$ in which [the queen]$_3$ [Atossa]$_3$ and [the ghost of [Darius]$_4$]$_5$ react to news of a military defeat; [that play]$_2$ is [the only classical tragedy on a contemporary, rather than mythical, subject]$_2$.	*All characters, authors, and works are annotated. Other mentions are ignored.*

Table 2.2: These examples from different datasets illustrate the differences in annotation standards, specifically for what is markable as a mention. Mentions are bracketed and entity clusters are subscripted with the same number. Table reproduced from Xia and Van Durme (2021).

shared task (Žabokrtský et al., 2022) aims to popularize a unified set of coreference annotations based on Universal Dependencies (Nivre et al., 2016) across 10 languages. In addition, some of these phenomenon in other languages, like zero anaphora, are present in specific domains in English, like recipes (Jiang et al., 2020).

Dataset$^{\text{lang.}}$	References	Domain	Training	Dev	Test
OntoNotes^{en}	Pradhan et al. (2012) and Weischedel et al. (2013)	Mixed written texts and transcripts	2,802	343	348
OntoNotes^{zh}			1,810	252	218
OntoNotes^{ar}			359	44	44
Semeval^{ca}	Recasens et al. (2010), Recasens and Martí (2010), Hoste and De Pauw (2006), Pradhan et al. (2007), Hinrichs et al. (2005), and Rodríguez et al. (2010)	Mixed written/spoken texts	829	142	167
Semeval^{es}			875	140	168
Semeval^{it}			80	17	46
Semeval^{nl}			145	23	72
Semeval^{de}			900	199	136
Semeval^{en}			229	39	85
RiddleCoref^{nl}	van Cranenburgh (2019)	Literature	23	5	5
ANCOR^{fr}	Muzerelle et al. (2014)	Spoken language	-	-	-
RuCor^{ru}	Toldova et al. (2014)	Mixed written texts	305	-	-
AnCor^{ru}	Budnikov et al. (2019)	Mixed written texts	268	0	127

Table 2.3: This table lists a couple of examples of non-English (collections of) datasets, along with their dataset statistics. The table is far from exhaustive. The statistics for ANCOR^{fr} and RuCor^{ru} may not be accurate as what is available and what is reported across papers sometimes differ. ANCOR^{fr} reports 500K tokens in total.

A third group is focused on using coreference resolution as diagnostics for models, typically focusing on pronoun resolution. For these (English) datasets, listed in

Table 2.4, researchers are typically more interested in probing knowledge or bias of models trained on other objectives, although there is some work on directly evaluating on and targeting sentence-level coreference (Kocijan et al., 2019). In this setting, the evaluation is more akin to a classification problem, as there is only one mention of interest in each example and ambiguous examples requiring editorial oversight are often excluded. Therefore, unlike the first and second groups, it is possible to compare a single model more fairly across multiple datasets, and evaluation is more straightforward, as accuracy or exact match F1 is reasonable. Beyond pronoun disambiguation, coreference resolution is also understood to be a core NLP task, and so there is also some work that includes coreference as part of a broader suite of probing tasks for pretrained language models (Tenney et al., 2019b; Tenney et al., 2019a).

Dataset	References	Probing phenomenon	Training	Dev	Test
WSC	Levesque et al. (2011) and Wang et al. (2019a)	Commonsense	554	104	146
DPR	Rahman and Ng (2012)	Commonsense	1,316	0	564
Winogender	Rudinger et al. (2018)	Gender bias	0	0	720
Winobias	Zhao et al. (2018)	Gender bias	0	1,580	1,580
GAP	Webster et al. (2018)	Gender bias	2,000	400	2,000
Edge probing	Tenney et al. (2019b)	Coreference	207,830	26,333	27,800

Table 2.4: Some examples of probing datasets. Most of the examples in these datasets are sentences or sentence pairs. For edge probing, each example targets a different mention, and so each sentence often occurs multiples in the dataset.

Finally, there are a few datasets for *cross-document* coreference resolution, such

as Cybulska and Vossen (2014, ECB), Cattan et al. (2021b, SciCo), and Ravenscroft et al. (2021, CD2CR). In this setting, rather than drawing mentions from just a single document D, mentions can be drawn from any document in a corpus, $D \in \mathcal{D}$, each coming from a different author or source. This can lead to more interesting clusters than within-document coreference, especially with regards to events and technical terminology, as independent authors may describe the same entity or event differently. This setting notably contrasts with single-document coreference resolution as the authors are not writing with the intention of references resolving across documents. Furthermore, only recently have larger-scale datasets been published, and only in English.

2.2.2 Annotation

There have been several interfaces used for coreference resolution annotation. The main interfaces used are the brat rapid annotation tool[8] (Stenetorp et al., 2012), MMAX2 (Müller and Strube, 2006), and SACR (Oberle, 2018). These tools are tailored for fast local (offline) expert annotation and designed for multiple layers of linguistic annotation.

Coreference resolution is challenging to annotate for several reasons. First, to ensure consistency, there are many rules and edge cases to learn (the OntoNotes coreference guidelines are 12 pages long and assume a linguistics background). It

[8] http://brat.nlplab.org

is also structured at the document-level, meaning each annotation requires more attention and cognitive load compared to many sentence-level tasks that can be crowdsourced at scale. Nonetheless, there has been recent work towards simpler interfaces for crowdsourcing coreference resolution (Yuan et al., 2021). Unlike previous interfaces, where users select the span and then link it to a previously selected span, Yuan et al. (2021) suggests spans to the user, as determined by another model (e.g. a syntactic parser or a different coreference system). The spans are subsequently merged during postprocessing (or not at all, in their setting for active learning). Crucially, by treating each span independently and precomputing a high-recall set of markables, the task is more easily crowdsourceable as the question asked from annotation is the simpler Q2 as opposed to Q1.

To measure annotation agreement (e.g. in the case of Pradhan et al. (2012, OntoNotes) or Chen et al. (2018, PreCo)), a subset is typically annotated with 2-way redundancy and then checked for mention boundary agreement (in some cases where mentions are already provided via an NER model or parser, this step can be skipped) and cluster agreement. For any disagreeing annotations, a third party can be brought in to adjudicate, resulting in a set of gold annotations. Then, the original annotations can be compared the gold clusters with the MUC metric, described in the next section. The 2-way redundant annotations can also be compared by the MUC metric.

2.3 Evaluation of coreference systems

There are a few standard metrics used to measure the accuracy of predictions for coreference resolution. However, since there are several dimensions (mention detection, mention linking, and mention clustering), there is no single, universal metric that absolutely measures the performance of a system. Many works also perform qualitative error analysis to identify the types of mistakes made by a model, which cannot be easily discerned from the quantitative metrics.

2.3.1 Quantitative Evaluation

Nonetheless, there are two commonly accepted quantitative metrics. The first measures the system's ability to perform mention detection. The second concerns the model's ability to perform coreference linking.

Mention Detection

The mention detection metric is calculated using the F1 score of the exact match of the span boundaries. In a system which first detects mentions before linking what is found, the *recall* of mentions is a more important metric to optimize, as the subsequent linking or clustering model can filter out false positives but would not rediscover false negatives.

CoNLL 2012 Clustering Metrics

There are three metrics for evaluating the links and clusters, and the standard approach suggested by the CoNLL 2012 shared task (Pradhan et al., 2012) is to report the unweighted average of the three metrics. Following prior convention, we let $K = \{k_1, ..., k_{C_k}\}$ be the set of C_k *keys* (i.e. gold) clusters and $R = \{r_1, ..., r_{C_r}\}$ be the set of C_r response (predicted) clusters. These metrics are not perfect (Moosavi and Strube, 2016); in fact, perhaps the only reason they continue to be used is because it is the easiest way to compare against prior work, such as those shown in Table 2.6.

The **MUC** (Vilain et al., 1995) metric aims to capture the accuracy of the predictions in terms of link edit distance, i.e. how many links need to be added or removed between R and K. For a cluster $k_i \in K$, let $p_R(k_i)$ be the set of clusters in R such that they "partition" k_i with elements not in R treated as singletons, i.e.

$$p_R(k_i) = \{r_j \in R \mid r_j \cap k_i \neq \emptyset\} \cup \{\{k_i\} \mid k_i \notin \bigcup_{r_j \in R} r_j\}. \tag{2.1}$$

Then, the MUC recall metric is defined as

$$MUC_{\text{recall}} = \sum_{k_i \in K} \frac{|k_i| - |p_R(k_i)|}{|k_i| - 1}. \tag{2.2}$$

As an example, if a key cluster $k_i = \{A, B, C, D\}$ were split into $r_1 = \{A, B\}, r_2 = \{C, D, E\}$ in the response, it would only take one link edit to merge r_1 and r_2, hence, the numerator is only one smaller than the denominator.

Furthermore, this metric only measures recall. Precision is computed by switching the key and response, i.e.

$$MUC_{\text{precision}} = \sum_{r_j \in R} \frac{|r_j| - |p_K(r_i)|}{|r_j| - 1}. \tag{2.3}$$

This metric alone has several flaws (Bagga and Baldwin, 1998; Luo, 2005; Cai and Strube, 2010; Moosavi and Strube, 2016). First, it incorrectly assigns importance to links: a split of a size-2 cluster is deemed more harmful than the split of the largest cluster, while an overmerge of two singletons is also more harmful to the metric than merging the two largest clusters. Yet, for most applications, we would expect predictions related to the largest clusters to be more critical. Second, it can be easily gamed by overmerging, as recall would be perfect and precision would only be slightly affected since there are not too many links that would need to be broken. Also, this metric does correctly account for singleton clusters as there are no links in singleton clusters. However, accurate scoring of singleton clusters is valuable for almost all datasets.

$\mathbf{B^3}$ (Bagga and Baldwin, 1998) address some of the problems of MUC by accounting for cluster size. Rather than computing an aggregate recall based on the number of links in each cluster of the key, this metric computes the precision and recall *per mention* of the key, which now better takes into account entity size. In particular, the *mention recall* for a particular mention m_i is $\frac{|R(m_i) \cap K(m_i)|}{|K(m_i)|}$, where $R(m_i)$ and $K(m_i)$ corresponds to the clusters containing m_i. Furthermore, this is averaged across all

mentions,

$$B^3_{\text{recall}} = \sum_{k_i \in K} \sum_{m_j \in k_i} \frac{1}{k_i} \frac{|R(m_i) \cap K(m_i)|}{|K(m_i)|} \tag{2.4}$$

$$= \frac{\sum_{k_i \in K} \sum_{r_j \in R} \sum_{m_j \in r_j \cap k_i} \frac{|r_i \cap k_j|}{|k_j|}}{\sum_{k_i \in K} |k_i|} \tag{2.5}$$

$$= \frac{\sum_{k_i \in K} \sum_{r_j \in R} \frac{|r_i \cap k_j|^2}{|k_j|}}{\sum_{k_i \in K} |k_i|} \tag{2.6}$$

This further rewards larger clusters. As a concrete example, suppose a cluster of size $2n$ could be split into two clusters of n each or into two clusters of $2n-1$ and 1. Under MUC, this would have the same recall of $\frac{2n-2}{2n-1}$. However, under B^3, there is substantially more error in the former split than the latter.[9] Like MUC, B^3 precision is computed by swapping K and R.

However, there are still some issues unresolved by B^3. Like MUC, if a response decides to merge all clusters into a single cluster (or treat them all separately as singletons), the metric will yield perfect recall or precision.

CEAF$_{\phi_4}$ (Luo, 2005) argues that the counterintuitive precision and recall from MUC and B^3 arise from the reliance on intersection between the key and response clusters. They address this by instead computing an alignment between the key and response clusters first by finding an optimal bipartite matching where edges are scored

[9] The former would have 0.5 while the latter would have a higher value of $1 - \frac{1}{n} + \frac{1}{2n^2}$.

using the Dice similarity function, $\phi_4(K_i, R_j) = \frac{2|K_i \cap R_j|}{|K_i|+|R_j|}$.[10]

After aligning each k to $g^*(k_i) \in R$, the recall can be computed by comparing against the optimal matching (scoring K with itself), and precision is computed by swapping K and R, as before.

$$CEAF_{\text{recall}} = \frac{\sum_{k_i \in K} \phi(k_i, g^*(k_i))}{\sum_{k_i \in K} \phi(k_i, k_i)} \tag{2.7}$$

One issue with this metric is that because each cluster is only aligned to one cluster, correct predictions in a second cluster may be ignored in the recall computation and penalized in the precision computation.

Other metrics

Other metrics have been proposed to further address the issues, although they have not been widely adopted. These include BLANC (Recasens and Hovy, 2011) and LEA (Moosavi and Strube, 2016), which offer some improvements around singletons and further balancing the importance of cluster sizes. However, with the CoNLL 2012 shared task implementing and releasing and official scorer averaging MUC, B^3, and $CEAF_e$ (Pradhan et al., 2012), most recent research in coreference resolution has reported on the average of these three metrics, and it has become the *de facto* standard metric used for comparing subsequent research. Despite the widespread adoption,

[10]This is also known as the *entity*-based CEAF. There is a less commonly-used *mention*-based CEAF with $\phi_3(K_i, R_j) = |K_i \cap R_j|$.

these metrics all depend on an exact boundary match, which may not generalize well to all languages.

2.3.2 Qualitative Analysis

In addition to quantitative metrics, researchers also perform qualitative analysis for learning about the errors made by their systems. Kummerfeld and Klein (2013) introduce an automated method to analyze the error types (over-merging, over-splitting, conflation). Subsequent work on coreference resolution has introduced their own set of qualitative analysis. For example, Lee et al. (2017) look at longer spans, head words, similar strings, and world knowledge, and Joshi et al. (2019) additionally analyze head words, pronouns, and document length. Generally, the findings are model-dependent, but doing this type of analysis helps us better understand where gains are coming from when they are obtained, such as when increasing model size.

2.4 Automatic systems

Coreference resolution was classically treated as a syntactic task (e.g. Hobbs (1978)), where the mention spans are provided along with the document and algorithms are used on top of syntactic parses to resolve references. Many approaches therefore saw boosts with the emergence of improved parsers. System development improved thanks to these parsers, along with richer features like curated word lists. As one example of

such system, I describe the multipass sieve (Raghunathan et al., 2010), which is the culmination of several rich rule and feature-based systems and still has an influence on models a decade later (Otmazgin et al., 2022). Then, I show how learned neural features have replaced and simplified several of the rules while still maintaining a pipeline-like system design.[11]

2.4.1 Sieve-based approach

The multi-pass sieve-based approach (Raghunathan et al., 2010) combines several rules, features, and constraints from prior work into a single system. By starting with high-precision rules (e.g. exact string match), the system makes easier decisions first. As the rules are relaxed, precision is traded off for recall, leaving a final set of predictions that is balanced between precision and recall. The final clusters are formed using the transitive closure of all pairwise links.

The rules take advantage of a variety of syntactic constraints and features, reproduced in Table 2.5, along with the effect of each additional pass on *pairwise* precision and recall. These passes take advantage of syntactic parses, head-finding rules, relative parse tree structure, NER labels, and other curated or mined lexical resources for acronyms, demonyms, stop words, gender, and animacy. One feature

[11]See Elango (2006), Clark and González-Brenes (2008), and Ng (2010) for more comprehensive surveys of methods and datasets at that time. In particular, there were many supervised models prior to the multipass sieve system rooted in machine learning models like SVMs or log-linear models. While they involved *learning*, they did not use *learned representations* (Luo et al., 2004; Rahman and Ng, 2009; Durrett et al., 2013), which I discuss in Section 2.4.2.

they exclude is using semantic head matching (Haghighi and Klein, 2009), which relies on additional mined semantic constraints. Each pass uses a subset of these features to generate rules for new links in the coreference clusters. As a result, each pass lowers precision while improving recall, and the full sieve results in a balance between both. One modern drawback is that the system cannot easily learn from *data*, and therefore is fully deterministic across datasets. Yet because of this determinism, it has predictable behavior across datasets.

2.4.2 Learned features

With advances in statistical modeling and neural methods, features used in coreference resolution shifted to being *learned* instead. This gives scoring or ranking functions between mentions access to continuous and trainable information in addition to binary (or discrete) rule-based features that were used previously, such as those in the multipass sieve. For example, a combination of features can be learned to optimize whether a text span is a mention and whether two mentions are coreferring (Wiseman et al., 2015). Similarly, several models have made used entity-level, or global, features that include features like cluster size or shape (Björkelund and Kuhn, 2014). These features can also be learned with neural models, resulting in entity representations (Wiseman et al., 2016; Clark and Manning, 2016b). Together, these improvements using learned representations improved the state-of-the-art by over 3 F1 points.

However, these models were still focused on the scoring or ranking of mention-pairs

Pass	Type	Features	P	R	Avg. F1
1	N	exact extent match	-	-	-
2	N, P	appositive | predicate nominative | role appositive | relative pronoun | acronym | demonym	97.5	42.6	58.6
3	N	cluster head match & word inclusion & compatible modifiers only & not-i-within-i	97.0	51.1	66.6
4	N	cluster head match & word inclusion & not-i-within-i	94.4	57.1	71.1
5	N	cluster head match & compatible modifiers only & not-i-within-i	93.6	58.3	71.5
6	N	relaxed cluster head match & word inclusion & not-i-within-i	92.4	59.1	72.0
7	P	pronoun match	85.9	74.2	79.6

Table 2.5: This table, reproduced from Raghunathan et al. (2010), shows what rules are used for each pass of the sieve and the average (MUC and B^3) precision, recall, and F1 on the ACE2004-ROTH-DEV set (Bengtson and Roth, 2008). "Type" refers to the types of antecendents (**P**ronominal vs. **N**ominal) that are targeted by that pass. The sieve targets initially high-precision rules and the later passes trade off precision for recall, ultimately boosting F1. In pass 2, two of the constraints need to be satisfied; these constraints are determined by syntactic parsers, acronym detection algorithm, an Wikipedia demonyms. not-i-within-i refers to one NP not being a child of another NP in the antecedent cluster. Pronouns are enforced by number, gender, person, animacy, and entity type agreement.

and cluster-pairs. Despite their claims that their models were end-to-end, they focus on the easier *linking* version of the task, where mention boundaries are provided by a different system. This setting is fair, and perhaps even common, for applications of coreference resolution (discussed later in Chapter 3). However, the inability to jointly learn the mention boundaries despite learned mention and entity representations led to the current generation of end-to-end neural models.

2.5 End-to-end neural models

In the previous section, the methods described all have multiple steps. Typically, they consist of mention detection, mention pair scoring or ranking, and subsequently mention-entity clustering or ranking. Mentions are detected through a syntactic parser (or rule-based system), mention pairs were scored using a mix of heuristics and learned features. Formation of the clusters varied based on approach: some approaches used a simple transitive closure while others used global features based on the full entity cluster.

Lee et al. (2017) combine all steps into a single end-to-end model (E2E). They introduce an approach which combines the first two steps by relying on text *span representations*, while they take the transitive closure to determine the full clusters. By directly learning high-dimensional span embeddings that are used both for mention detection and linking, this approach was found to be more effective than the pipeline

approach. This foundational model in discussed in more detail in Section 2.5.1.

With the introduction of the end-to-end model, research on neural coreference models has diverged into several directions as to improve the model's technical foundations, address some architectural limitations, and "catch up" to relevant pre-neural work. Section 2.5.2 discusses 1) extensions of the original model by improving the mention detection, linking, and clustering algorithms; 2) limitations of the end-to-end model adaptations to accommodate larger inputs; 3) an approach that revisits the notion of "span" and argues instead for word-level (and therefore smaller) embedding sizes.

2.5.1 End-to-end (e2e) model

Lee et al. (2017) introduced the first end-to-end model (E2E) for coreference resolution by jointly detecting mentions and coreference with a single objective. For an input document D, the task is formulated as labeling each text span with either a preceding text span or a non-span "dummy" label which indicates that the span is not a mention. Doing so is sufficient to recover the full coreference clusters by using a transitive closure between the predicted links. This process is illustrated in Figure 2.1 (reproduced from Lee et al. (2017)).

Formally, the model considers all spans up to a certain length, T, such that each span is fully contained within a sentence. There are no additional syntactic constraints on these spans. For each span i, the set of possible antecedents are all preceding spans,

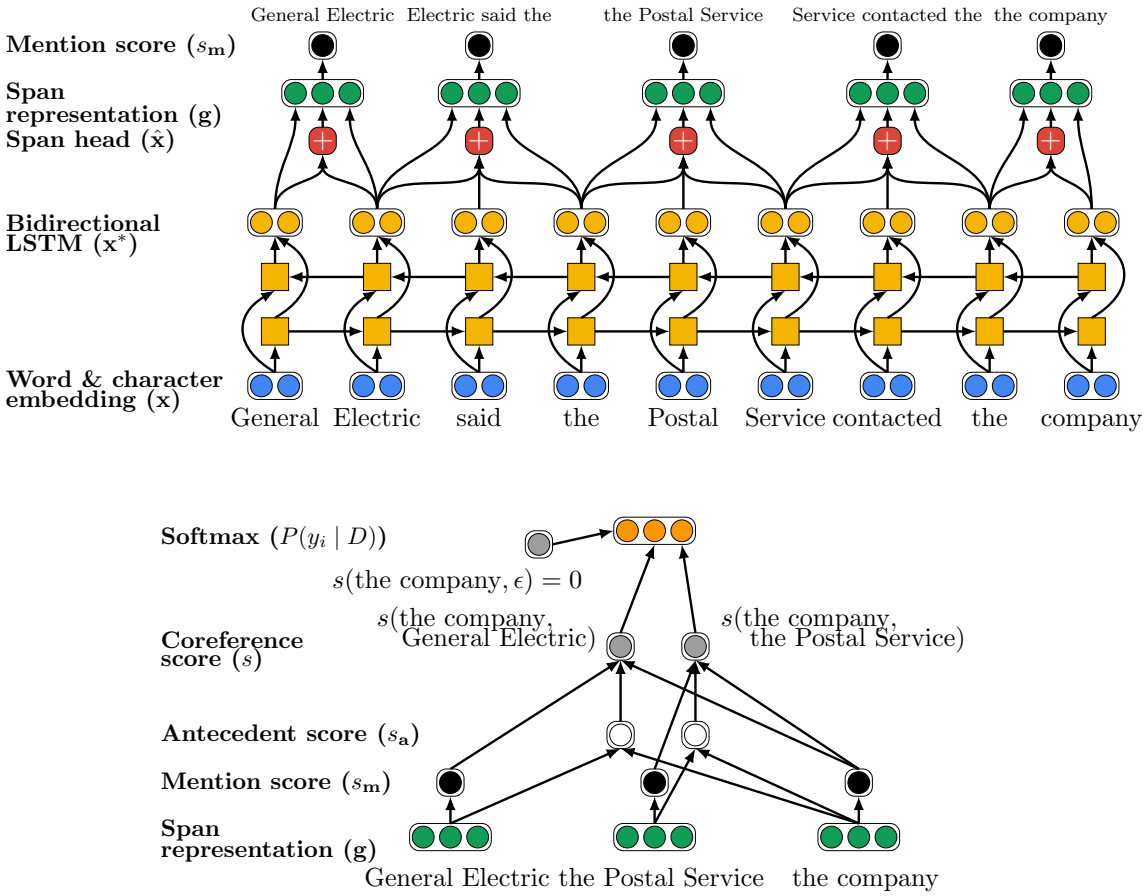


Figure 2.1: These figures, reproduced from Lee et al. (2017), show the overall framework of end-to-end neural coreference systems. First (top), text is encoded into span representations, in this case, LSTMs are used. Next (bottom), span representations are scored pairwise and against a dummy ϵ "span" which receives a fixed score of 0. For each span representation, the best-scoring preceding span is its label.

which is denoted as $\mathcal{Y}(i) = \{\epsilon, 0, 1, ..., i-1\}$ where ϵ is "dummy" label. The goal is to correctly predict the assignments y_i, i.e. maximize the conditional probability of $P(y_1, ...y_N \mid D)$, which can be decomposed as

$$\prod_{i=1}^{N} \frac{\exp(s(i, y_i))}{\sum_{y' \in \mathcal{Y}(i)} \exp(s(i, y'))} \tag{2.8}$$

The scoring function is further decomposed to include a *mention* scoring function, s_m, and a *pairwise* scoring function, s_a. For non-dummy spans $s(i,j) = s_m(i) + s_m(j) + s_a(i,j)$ while $s(i,\epsilon) = 0$.

To score spans, the E2E model creates contextualized span and token representations. To do so, each token is first embedded with the concatenation of two word embeddings (Pennington et al., 2014; Turian et al., 2010), and a bidirectional LSTM is used at the sentence level.[12] Then, span representations are formed by using an attention weighted "head" mechanism, and the final representation $\mathbf{g}_i = [\mathbf{x}_{\text{start}}, \mathbf{x}_{\text{end}}, \mathbf{x}_{\text{head}}, \phi(i)]$ where ϕ is an embedding based on the size of i.

With span representations defined, both s_m and s_a can be defined,

$$s_m(i) = \mathbf{W_m}\text{FFNN}_m(\mathbf{g}_i) \tag{2.9}$$

$$s_a(i,j) = \mathbf{W_a}\text{FFNN}_a([\mathbf{g}_i, \mathbf{g}_j, \mathbf{g_i} \circ \mathbf{g_j}, \phi(i,j)]), \tag{2.10}$$

where $\phi(i,j)$ is a feature function capturing the distance between the two spans, the document genre (of OntoNotes), and whether i and j were spoken by the same speaker.

To train the model, the log-likelihood of the antecedents for each span is maximized, Because there are multiple correct antecedents, the probability of *all* antecedents are maximized.

[12]Later adaptations of the E2E model would instead use contextualized encoders, typically a pretrained language model, like Peters et al. (2018).

$$L_{\text{coref}} = -\log \prod_{i=1}^{N} \sum_{\hat{y} \in \mathcal{Y}(i) \cap \text{Gold}(i)} P(\hat{y}) \tag{2.11}$$

Because the objective is fully differentiable, all parameters can be trained with backpropagation. Note that the mention scorer is implicitly learned as it receives training signal from the final objective; there is no auxiliary signal to encourage "good" mentions or heads.

Finally, some of the hyperparameters are carefully chosen to keep the model tractable on a single GPU. First, only spans up to a certain length, L, are considered initially. Second, these spans are pruned to keep the top $\lambda|D|$ total spans (where $\lambda = 0.4$), and finally, only the most recent K antecedents are considered. Implicitly, this encodes some biases about the size of spans, frequency of mentions, and selectional constraints.

As highlighted in Table 2.6, this approach substantially outperformed previous methods. However, there were still several limitations of this work, which subsequent work explored:

1. Addressing the artificial constraints used to keep the model tractable on a single GPU.

2. Better methods (than transitive closure) for decoding pairwise scores into clusters.

3. What happened to the semantic and syntactic agreement constraints that were relied on in prior work?

4. Extensions of the model to datasets with singleton clusters.

2.5.2 Variations of neural end-to-end models

Due to the success of the new model and approach, the E2E model raised several questions that were answered by subsequent work, leading up to the present day. This section discusses several of the variations.

Using a coarse-to-fine pairwise scorer

In OntoNotes, there are spans of text that are longer than the maximum length of $L = 10$ specified in the E2E model. However, increasing the length of spans under consideration also increases the total number of spans, which causes the number of candidate antecedents per span, K, to be reduced. In the E2E model, K is the determined to be the most recent preceding spans. By reducing K, the model would limit the distance between mentions. Lee et al. (2018) propose an adjustment by using a *coarse* scorer to determine K. Instead of selecting K by recency, a bilinear *coarse* scorer, parameterized by the matrix $\mathbf{W}_c$, is learned to inexpensively score between the embeddings for spans i and j: $s_c(i, j) = \mathbf{g}_i^\top \mathbf{W}_c \mathbf{g}_j$. The top-$K$ scoring spans j are chosen for each span i. These are then used by the *fine* scoring function, $s_a(i, j)$ from E2E.

"Their heads are gone, if it please your Majesty!" [**the soldiers**] shouted in reply.
"That's right!" shouted the Queen. "Can [**you**] play croquet?"
[**The soldiers**] were silent, and looked at [**Alice**], as the question was evidently meant for [**her**].
"Yes!" shouted [**Alice**].

Figure 2.2: In this dialogue from Caroll's *Alice's Adventures in Wonderland*, the pairwise score of (**the soldiers**, **you**) is high, as that is who the Queen is speaking to. In addition, because Alice responds, the score of (**you**, **Alice**) is also high. However, (**the soldiers**, **Alice**) would get a low score. Higher-order inference aims to resolve this puzzle.

Higher order inference

One of the types of mistakes that appeared preventable in the predictions made by the E2E model was inconsistent clusters. Consider the example in Figure 2.2, two pairwise predictions (A, B) and (B, C) may score positively given the local context, but (A, C) would score negatively. As the cluster decoding is naively performed greedily based on the pairwise scores (by taking argmax for each span), it is possible to erroneously merge (A, C). Given the already-computed pairwise scores, it should be possible to avoid these errors.

Lee et al. (2018) propose higher-order inference to address the issue of global coherence. They introduce a *span refinement* method which updates $\mathbf{g}_i$ given its current representation and the representation of its attended antecedent, which is the weighted sum of antecedent embeddings by their probabilities, $P(y)$. Kantor and Globerson (2019) simplify the process further by treating each embedding as the sum of its antecedents. Xu and Choi (2020) propose clustering methods of span refinement where the mention representation is updated to be an interpolation between the

original embedding and an attended entity embedding. Span refinement is a process that can be iterated, resulting in updated representations $\mathbf{g}_i^n$ for each span i on iteration n.

However, Xu and Choi (2020) also find that higher-order inference may have a fairly small effect in resolving the inconsistencies, as many of them appear to be resolved through improved contextualized representations. It is also possible that the hardest instances presented, like in Figure 2.2, are rare in OntoNotes 5.0 but more common in other domains or languages.[13]

Contextualized embeddings and neural scorers

In line with the adoption of pretrained language models for encoding text (see survey by myself and coauthors (Xia et al., 2020b)), improved contextualized text representations have also boosted performance of coreference resolution models and are one of the most significant contributors to performance gains in the last several years. By replacing the word embeddings and sentence-level LSTM with ELMo (Lee et al., 2018), BERT (Joshi et al., 2019), SpanBERT (Joshi et al., 2020), and LongFormer (Beltagy et al., 2020), performance on OntoNotes 5.0 has improved significantly despite minimal changes in model architecture. In particular, the LongFormer architecture

[13]While cases like Figure 2.2 are rare and occasionally intentionally ambiguous, e.g. to build suspense, systems that perform poorly and over-rely on proximity, exact string match, and other person or gender heuristics may even fail on example (3) from Section 2.1. Even a few errors of this type can have a large impact, especially for large clusters or in longer texts. In this example, if **her**$_{\text{SUMIRE?}}$ incorrectly selects **Miu**, then the subsequent **her**$_{\text{SUMIRE}}$ and all future instances may become conflated with **Miu**, essentially merging two clusters that should be distinct.

has allowed the models to tractably handle longer documents, like LitBank, without artificially limiting the context size (Toshniwal et al., 2020b). Part of the strength of the larger pretrained models comes from their pretraining strategies, which consumes text that is orders of magnitude larger than what is available in coreference corpora. This leads to gains in difficult semantic situations, where world knowledge might be useful.

Another area of improvement is the neural scorer. In the E2E model, the pairwise scoring function is fairly simple, consisting of a single network that takes as input the concatenation of the start embedding, end embedding, an attention-weighted head word, and relatively simple features. Wu et al. (2020) reframed the coreference resolution task as a question-answering (QA) task. In particular, to compute the pairwise probability $P(i \rightarrow j)$, they predict the probability span j is the answer to a query regarding i. This is a considerably more expensive step as it requires rules to set up the query and expensive memory usage and inference time to compute the scores. However, it yields better performance and is the best-performing model on OntoNotes 5.0 to date. Furthermore, this model uses auxiliary QA data for additional training. Nonetheless, due to its cost and difficulty of reproducibility, the QA-style neural scorer has not been an active area of research.

Singletons

The field of coreference resolution, around 2013-2020, has over-relied on a single dataset: OntoNotes. While OntoNotes 5.0 covers multiple languages and domains, its fundamental flaw is that singletons, or clusters of size 1, are not annotated. This means that the models that are developed, like E2E, are incapable of predicting singletons. This was less an issue when mention detection and clustering were pipelined, as the unlinked mentions could be treated as singletons. However, with end-to-end models, it is less clear that "high-scoring" unlinked mentions are necessarily singletons, and in fact I explore this further in Chapter 3.

Several works attempt to address this issue by including mention classification as an auxiliary objective and observe that it results in minor improvements (Zhang et al., 2018a; Swayamdipta et al., 2018). In particular, if $\mathcal{M}$ is the set of mentions kept by the mention pruning step and m_i^* is the correct label for m_i (either it is a valid mention (1) or it is not (0)), we can write the probability that m_i is correctly predicted as,

$$P(m_i^*) = m_i^*(\text{sigmoid}(s_m(i))) + (1 - m_i^*)(1 - \text{sigmoid}(s_m(i))) \tag{2.12}$$

Then, the new loss resembles

$$L_{\text{joint}} = \alpha_c L_{\text{coref}} + \alpha_m L_{\text{mentions}} \tag{2.13}$$

$$= -\alpha_c \log \prod_{i=1}^{N} \sum_{\hat{y} \in \mathcal{Y}(i) \cap \text{Gold}(i)} P(\hat{y}) - \alpha_m \log \prod_{m_i \in \mathcal{M}} P(m_i^*), \tag{2.14}$$

for some tuneable hyperparameters α_c, α_m. However, this method was initially only investigated with the OntoNotes 5.0 dataset. Subsequent work has explored this type of auxiliary objective for other datasets and it is now a standard way of detecting and separately predicting singleton clusters (Xu and Choi, 2020; Toshniwal et al., 2020a; Toshniwal et al., 2021; Xia and Van Durme, 2021). In Section 4.2, a more rigorous treatment of singletons, in the context of the model used throughout this thesis, is presented.

Incremental models

One limitation not addressed by the prior models is the case where documents are too long. This leads to two issues: 1) The encoder cannot use the full context to encode tokens; and 2) As n increases, the memory usage of the model also increases linearly. Both of these issues can be addressed by taking an incremental approach (Webster and Curran, 2014). In the incremental approach, entity clusters are formed while the model is processing each sentence. They argue that this is psycholinguistically motivated, as reading order in discourse is typically sufficient to disambiguate references. Concretely,

the method assumes a set of named entities and noun phrases extracted from the document. Then, for each mention, the algorithm finds the best existing cluster (including a **new** cluster) that it should link to. Their scoring function between mention and cluster is determined by rich discourse, like syntax and semantics (Raghunathan et al., 2010), and stack features, like depth.

This approach has been neuralized in an end-to-end manner where the mentions are detected using the mechanism from the E2E model and the mention-cluster pairs are scored via a learned neural scorer (Xia et al., 2020a; Toshniwal et al., 2020b; Yu et al., 2020b). This model is discussed in more detail in Chapter 4. The result of this approach is that the model can accommodate much longer, book-length, documents in constant memory and subsequently, has led to improvements on datasets with longer documents, like LitBank. There has also been recent extensions to the *online* setting, which is common in dialogue (Xu and Choi, 2022; Xia and Van Durme, 2022).

Revisiting Heuristics

As contextualized models have improved, there has been a resurgence of word-level models that revisit some of the pre-neural heuristics. Lee et al. (2017, E2E) softened the notion of "head words" within each mention span by using an attention-weighted span representation. This contrasts with prior work, such as Bengtson and Roth (2008), which found success in first predicting head words, before subsequently computing the extent of the syntactic span. Many of the features used in that system hinged on the

head word. While the neural span-based methods were initially better, recent work has revisited the idea of word-level models. For example, Dobrovolskii (2021) takes an approach almost identical to Bengtson and Roth (2008), except it uses modern neural models and pretrained language models. Kirstain et al. (2021) place importance only on the span boundary tokens rather than the full span. These models both discard the attention-weighted span representation, and as a result, these word-based approaches significantly reduce the size of a mention representation. They both demonstrate superior performance to the comparable span-based models.

Recently, Jiang and Cohn (2021) and Jiang and Cohn (2022) have found that providing the models with additional syntax or semantic supervision via parse trees and semantic roles during training can lead to additional improvements, mirroring prior work on joint modeling, like Durrett and Klein (2014). The drawback to these methods is the reliance on gold syntactic and semantic supervision at training time. This is rarely available, and it is unclear whether the gains observed transfer to new domains.

2.6 Remarks

What is state-of-the-art?

Table 2.6 shows reported claims for state-of-the-art models over the last several years. Currently, the state-of-the-art model is Wu et al. (2020), reporting a score

of 83.1 on OntoNotes 5.0. A standing record for over 2 years is unusual in the modern NLP landscape. However, due to the cost of that model, most research in coreference resolution iterates on the coarse-to-fine (C2F) model with SpanBERT encoder (Lee et al., 2018; Joshi et al., 2020; Xu and Choi, 2020). These have yielded small improvements as a result of better features and heuristics, or perhaps better hyperparameter search and engineering, and closing the gap while maintaining relatively light, single-GPU models. There are now multiple models that feature improvements over the C2F SpanBERT model, yet no attempt to assemble the lessons learned from each method into a single model. Are the gains derived from these works complementary or overlapping?

The original OntoNotes dataset is two-way annotated, although the report only mentions inter-annotator MUC score for various genres. Meanwhile, the MUC score of the best systems are beginning to approach those agreement scores, and so another area to explore is a better understanding of the human baselines and how they compare to model predictions.

While work continues on OntoNotes 5.0 and closes the gap towards Wu et al. (2020) and human baselines, the focus in this thesis is on *efficient* and practical approaches for applying coreference resolution as part of a larger system. To this end, improving the state-of-the-art is not the focus; catching up on what was left behind is. Chapter 3 describes a few case studies motivating the need for efficient methods.

Model (contribution)	Type	MUC			B^3			$CEAF_{\phi_4}$			
		P	R	F1	P	R	F1	P	R	F1	Avg. F1
Pradhan et al. (2012)	human	83.2-96.0	-	-	-	-	-	-	-	-	-
Jiang and Cohn (2022)	syntax	87.3	87.1	87.2	81.1	80.9	81.0	78.8	77.2	78.0	82.1
Jiang and Cohn (2021)	syntax	87.2	86.7	87.0	81.1	80.5	80.8	78.6	77.0	77.8	81.8
Dobrovolskii (2021)	word	84.9	87.9	86.3	77.4	82.6	79.9	76.1	77.1	76.6	81.0
Kirstain et al. (2021)	word	86.5	85.1	85.8	80.3	77.9	79.1	76.8	75.4	76.1	80.3
Wu et al. (2020)	QA	88.6	87.4	88.0	82.4	82.0	82.2	79.9	78.3	79.1	**83.1**
Xu and Choi (2020)	hoi	85.9	85.5	85.7	79.0	78.9	79.0	76.7	75.2	75.9	80.2
Joshi et al. (2020)	PLM	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	79.6
Joshi et al. (2019)	PLM	84.6	82.4	83.5	76.5	74.0	75.3	74.1	69.8	71.9	76.9
Kantor and Globerson (2019)	hoi	82.6	84.1	83.4	73.3	76.2	74.7	72.4	71.1	71.8	76.6
Lee et al. (2018)	hoi, PLM	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
Lee et al. (2017)	e2e	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
Clark and Manning (2016b)	LF	79.9	69.3	74.2	71.0	56.5	63.0	63.8	54.3	58.7	65.3
Wiseman et al. (2016)	LF	77.5	69.8	73.4	66.8	57.0	61.5	62.1	53.9	57.7	64.2

Table 2.6: Claims of state of the art results on OntoNotes 5.0 (which is the CoNLL-2012 shared task), starting around 2015 with a few pipeline models followed by the E2E models and several of its variations. LF = learned features; e2e = end-to-end; hoi = higher order inference; PLM = pretrained language models or encoders; QA = question answering reformulation; word = word-based models; syntax = incorporating gold syntactic annotations at training.

Imprecise task definition

As discussed in Section 2.1, there are many definitions for the coreference task depending on the specific phenomena of interest. However, the most common benchmark is OntoNotes, which is also one of the harder-to-use English datasets because it lacks singletons. As a result, models trained on OntoNotes are good at answering "which spans corefer?" but perform poorly at the task to completely "find all the entities." The mismatch leads to inconsistent datasets and part of the contribution of this work is to provide methods to unify and leverage multiple, different datasets, as discussed in Chapter 5.

Pipeline and end-to-end approaches

Among the contributions of the E2E model is its decision to omit the use of a syntactic parser or NER model. In fact, they find that using parser-predicted spans hurts performance. This follows a general trend within NLP of favoring fully end-to-end approaches. While there are some efforts in bringing back syntax into coreference resolution models, until recently, they have only seen limited success (Swayamdipta et al., 2018; Jiang and Cohn, 2021). Furthermore, this has not led to the return to pipeline systems for achieving state of the art performance on OntoNotes.

On the other hand, taking a pipeline approach is still a core component of proposing mentions for annotation purposes, as manually selecting span boundaries is a fairly

expensive process, and a pipeline approach can be better tuned to fit the specifications of the downstream task. Furthermore, Chapter 3 discusses downstream applications consisting of model pipelines, where another system is first run to extract mentions (e.g. semantic arguments, protein names, literary characters) and for those settings, we should still be interested in the pipeline setting by reporting metrics with gold mentions or disentangling mention detection and linking (Wu and Gardner, 2021).

As a probe for language understanding

Coreference resolution has also been included as a standard "task" in several probing benchmarks for language models and language understanding, as highlighted in Table 2.4. While many NLU challenges can be accurately cast into coreference resolution, they are not representative of the natural distribution of coreference. Furthermore, these datasets usually assume or provide mention boundaries are used for binary classification. Thus, from the lens of probing, coreference resolution is viewed as a semantic task, and challenges lie primarily in real-world understanding (Tenney et al., 2019b). Researchers have used these datasets to draw conclusions about pretrained language models. They find that the more recent, larger language models have an improved ability for common sense reasoning, under the Winograd Challenge, while they are also appear better at avoiding gender biases (Tenney et al., 2019a; Wang et al., 2019b; Raffel et al., 2020; Brown et al., 2020a). On the other hand, from the perspective of coreference resolution, the probing benchmarks are artificial

and rarely studied or used to evaluate coreference resolution models, with GAP being the primary dataset that is explored (Joshi et al., 2019; Kirstain et al., 2021).

Cross-document coreference resolution

While the scope of this dissertation does not extend to cross-document coreference resolution, this task, at the surface, shares many similarities with within-document coreference resolution described in this chapter. This section briefly contrasts the methods and challenges between cross-document coreference resolution and single-document coreference resolution; Bugert et al. (2021) gives a more comprehensive overview of recent methods and datasets.

In particular, many underlying models extend from the E2E model in terms of mention detection, scoring, and clustering. Furthermore, some of these models have also recently transitioned to sequential or incremental models, similarly citing efficiency concerns (Allaway et al., 2021). Furthermore, they have also seen significant improvement in performance when switching to a cross-document pretrained language model (Cattan et al., 2021a).

A major difference between the two tasks is how the dataset is typically annotated. The datasets used for cross-document coreference resolution are grouped by topic and subtopic. While the mix of topics aims to make topic matching harder, simple clustering methods in preprocessing can completely disentangle the topic clusters and coreference resolution can subsequently be performed between document pairs

with known shared entities or events. Another difference is that not all mentions are annotated in the cross-document dataset, pronouns are rarer while events are typically annotated. This means that without gold mention boundaries, prediction on these datasets are inconsistent and incomparable. Neural cross-document coreference resolution, especially datasets to support it, in the academic setting still lags behind those of within-document coreference resolution, despite the wide potential industrial applications.

Chapter 3

Motivation: Coreference resolution in practice

This chapter describes two case studies. One (Section 3.1) is a mildly negative result regarding pipeline coreference resolution models, where separating mention detection from coreference linking and training a pipeline hurts performance. On the other hand, Section 3.2 describes several real-world applications that precisely require a pipeline system due to domain or language changes. These studies motivate the remainder of the thesis, especially looking towards *using* coreference resolution models in practice.

3.1 Recovering Singletons

Note

This section discusses an attempt to reconstruct singleton clusters from OntoNotes 5.0, with the goal of showing that doing so will improve performance on OntoNotes 5.0 itself. This study was conducted in 2018, and so the state-of-the-art model at the time was by Lee et al. (2018) with ELMo (Peters et al., 2018) or BERT (Devlin et al., 2019). This model is described in more detail in Section 2.5.2. The work was written up as "**Singletons Matter: Complete Span-based Entity Mention Detection and Coreference Resolution**",[1] although it was never published due to the relatively weak empirical results. Certainly, it was not as strong as our later

[1]This work was performed with Ryan Culkin and advised by Benjamin Van Durme. Ryan worked on constructing the recovered spans while Ben advised the project.

work in Toshniwal et al. (2021), where we again reconstructed singleton clusters and named them "pseudo-singletons." In that work, we found that *training* with additional (*predicted*) (pseudo-)singletons can improve generalizability across datasets and also lead to a 1.1 F1 improvement on OntoNotes.

3.1.1 Abstract

This study presents a span-based neural model for noun-like mention detection and coreference resolution. One component directly optimizes for detecting mentions by augmenting training data with additional, syntactically derived singleton mentions from gold annotations, while another is trained on coreference linking. The pipeline performs comparably to the state of the art on OntoNotes, while outperforming it in predicting these mentions. While coreference resolution is fundamentally a clustering task, practical uses of models for the task desire high recall predictions of all mentions, *including singletons*, that can be referred to.

3.1.2 Background

As discussed in Section 2.1, in-document coreference resolution is the task of linking textual mentions of the same entity or event to each other within the same discourse. In the pipelined approach (Section 2.4), the mentions are first discovered (Raghunathan et al., 2010; Durrett and Klein, 2013) and subsequently linked or

(1): **Hong Kong Wetland Park, which is currently under construction**, is also one of **the designated new projects of the Hong Kong for advancing the tourism industry**.
(2A): **This** is a park intimately connected with nature, being built by the Hong Kong government ...
(2B): Hong Kong Disneyland, opening next year, is yet another one of **these projects**.

Figure 3.1: (1) contains two entity mentions, either of which could reasonably be referred to later in the discourse: (2A,B) are both plausible continuations. In OntoNotes, (2A) is the true subsequent sentence, and thus the **orange** mentions are both annotated, while the **violet** mention in (1) is a *singleton* referrent, and therefore unannotated in the dataset.

clustered using pairwise scorers and global features (Wiseman et al., 2016; Clark and Manning, 2016a; Clark and Manning, 2016b). These steps can be jointly combined in an end-to-end model which ranks candidate mentions and links them (Lee et al., 2017). Since a mention without links, or a *singleton* mention or cluster, is not related to any other mention in the text, they are ignored in the most commonly-used dataset (OntoNotes) and even in one of the metrics (MUC) for the task.

For systems relying on coreference resolution predictions, the ignored singleton clusters (mentions) are useful as they can be linked to either future mentions that the system has yet to see (if the data is streaming)[2] or to known entities from a knowledge base. Figure 3.1 shows an example of two entities, ***the park*** and ***the projects***, that a downstream system may need to track. However, coreference resolution datasets, like OntoNotes, are not typically exhaustively annotated for mentions. In the example, only one span of ***the park*** is annotated in the data, leaving ***the projects*** ignored or possibly treated as negative example of a span. This leads to models that are

[2]Chapter 4 is focused exactly on this streaming scenario.

trained on weak signal and partial annotations that are unable to use or predict these singleton mentions, if needed later. As a result, coreference models can have either low precision in matching syntactic constituents or low recall in extracting referable mentions, depending on how spans are extracted.

OntoNotes 5.0 coreference resolution annotations

The most commonly used and widely annotated benchmark for document coreference resolution is OntoNotes 5.0 (Pradhan et al., 2013; Weischedel et al., 2013), which consists of documents spanning multiple genres: news broadcasts, newswire, phone calls, and religious texts. Further, it is heavily annotated with gold syntactic parses, named entity types, coreference clusters, word senses and propositions. These rich annotations and size of the dataset[3] make it an attractive choice for training information extraction systems.

However, systems trained on OntoNotes will obtain the idiosyncrasies present in OntoNotes. In particular, only coreference links are annotated and so not all mentions are labeled (Figure 3.1). On the other hand, high recall systems for knowledge base generation or entity linking will expect exhaustive annotation of entities from the document. While it is possible to use rule-based systems to generate the mentions (Raghunathan et al., 2010; Durrett and Klein, 2013), surprisingly, they perform worse than the nonexhaustive annotations when used to train an end-to-end neural

[3]The English training, development, and test splits contain 2802, 343, and 348 annotated documents respectively.

coreference resolution system (Lee et al., 2017).

Span-based neural models

Vector-based span representations gained attention due to their contribution to performance in models for coreference resolution (Lee et al., 2017), semantic role labeling (He et al., 2018), semantic parsing (Peng et al., 2018), and knowledge graph extraction (Luan et al., 2018), and probing (Tenney et al., 2019b).

Most closely related to this work, Swayamdipta et al. (2018) effectively make use of syntactic annotations for both semantic role labeling and coreference resolution in a non-ELMo setting. However, they do not report the performance on the auxiliary span classification task or whether the knowledge was still retained by the end of training. Relative to that work, this study asks whether it is possible to for a model to retain its syntactic scaffold at inference.

Mention detection

Mention detection, along with named entity detection, is often presented as sequence tagging problem with a neural CRF-based tagger (Lample et al., 2016). Xu et al. (2017) addressed the case where named entities (and noun-like mentions) are nested (e.g. *"[University of [[Toronto]]")* by using a span-based approach and showed that it was competitive with sequence labeling techniques.

This study also takes a span-based approach for both exhaustively recovering

unannotated mentions because of the nested nature which can also occur in tasks like coreference resolution. For this task specifically, ***Hong Kong***, ***Hong Kong Wetland Park***, and ***Hong Kong Wetland Park, which is currently under construction*** from Figure 3.1 are all referable by pronouns. Downstream systems such as entity linkers need the ability to treat each of these mentions separately, and completeness is valuable. Section 3.2 discusses a system for which this was the case (Chen et al., 2019).

Baseline model

The baseline model is the C2F model described in Section 2.5.2 (Lee et al., 2017; Lee et al., 2018). Recall that this which consists of a two-step beam search. First all spans are scored and ranked, keeping the top k. Next, spans are scored pairwise and the top c possible antecedents for each of the k spans are re-scored. Notably, the learning objective does not assign loss to mentions that are not in the top k or make a distinction between singletons and non-mentions.

Zhang et al. (2018a) attempts to address this issue with a binary cross-entropy loss for gold mention detection. However, this objective still incorrectly penalizes singleton mentions. Furthermore, analysis of the baseline model showed that longer spans, which are rare in data, are also less frequently proper syntactic constituents than shorter spans (Lee et al., 2017). Both the scarcity of long spans and incorrect penalties in training objective motivate an exhaustive span-based approach for computing a

loss for all noun-like mentions rather than just gold mentions.

3.1.3 Augmenting Mention Annotations

Since the OntoNotes dataset is not annotated for singleton mentions, we develop a simple method to automatically extract noun-like expressions from several OntoNotes annotation layers, effectively augmenting OntoNotes with the missing singleton mentions. In this procedure, we extract all the noun phrases (NPs) and possessive pronouns[4] from the gold constituency parses, extract named entities from the named entity annotation layer, and then union the resulting recovered spans (*r-spans*). Note that in other works, noun phrases and possessives are always considered markable even if they are non-referential (Pradhan et al., 2007; Poesio and Artstein, 2008). While using gold named entities is unrealistic for most datasets, it is not far from other work which also assumes gold non-syntactic information, like semantic roles, at training (Jiang and Cohn, 2021).

Table 3.1 reports precision and recall for several r-span configurations; the final configuration achieves 97.73% recall and 32.24% of OntoNotes coreference cluster spans using an exact-match scoring protocol. The high recall functions as a sanity check – this method recovers virtually all of the non-singleton spans – while the low precision

[4]Other phrase- and word-level tag types (e.g. singular or plural noun, wh-pronouns, etc.) were found to almost always (1) overlap with an identical NP span or (2) overgenerate – e.g. For "*the hype*" (a NP), "*hype*" should not be included; NP is a reasonable level of abstraction that picks out text spans likely to be referred to by a pronoun. We include possessive pronouns because they are largely disjoint from NPs.

Configuration	P.	R.
NPs only	32.21%	86.83%
+ Possessive pronouns	**34.16**%	95.34%
+ Named entities	32.24%	**97.73**%

Table 3.1: Precision and recall of several r-span configurations treating OntoNotes coreference cluster spans as ground truth.

indicates that we recover missing singleton spans. Upon manual inspection, virtually all of the false negatives (2.27%) are events, which is expected since verb-like phrases are not extracted even though they are present in the dataset. The false positives are mentions by definition; they are either NPs, possessive pronouns, or named entities, all of which should be considered markable and belonging to a singleton.[5]

Previous work (Haghighi and Klein, 2010; Kummerfeld et al., 2011) has addressed the issue of missing singletons and made use of a similar procedure to recover missing singleton spans, with some important differences. First, where our approach considers all NPs (including nested NPs), prior work only considers the maximal NP projection. For example, given "*The government in Hong Kong*", we would recover "*The government*", "*Hong Kong*", and "*The government in Hong Kong*", whereas prior work would only recover the single maximal phrase. This is needed to be complete in collecting mentions.

Second, we include gold named entity annotations because they capture additional noun-like expressions that are not represented in the constituency parses, perhaps

[5]In retrospect, these could also contain *non-referential* NPs, and one flaw of this recipe is that these were not identified or filtered those out.

owing to quirks or errors in the annotations. Without named entity spans, i.e. with only the spans derived from the constituency parses, we recall 95.34% of the OntoNotes coreference cluster spans; adding named entity spans increases recall 2.39 percentage points to 97.73%.

Finally, whereas previous work in pipeline models use the Berkeley parser (Petrov et al., 2006) to obtain automatic parses and then used those as the base for extraction, we used gold constituency parses, so the resulting extracted spans are of higher quality.

3.1.4 Experiments and Results

The augmented dataset can be used in several experiments. The baseline end-to-end model is from Lee et al. (2018), discussed in Section 3.1.2. To accommodate the mention detection task, a binary cross-entropy loss was used in a multitask setting on all possible spans, following Zhang et al. (2018a). We sample each task at a rate λ_{task}.[6] To extract mentions, including singletons, from the baseline model, the scores of the top spans in the first beam are used. Both a fixed score threshold and a span-ranking approach was attempted to determine the top k candidate mentions to be considered for clustering. For training, a burn-in period of 20K iterations on mention detection was applied to reduce early stage memory issues with a low score threshold, though both the score threshold and burn-in did not have significant effects on performance on the development set. Finally, I also train a baseline linking-only model purely on

[6] $\lambda_{\text{mention}} = 0.05, \lambda_{\text{coref.}} = 0.95$ was determined to work best after a small search. Note that this contrasts against subsequent work which use equal coefficients.

Model	**r-spans**			**Gold**	
	P.	R.	F1	P.	R.
Baseline	66.8	73.8	70.1	28.4	97.0
Multitask	81.4	**96.4**	88.3	26.6	**97.4**
MD	**95.1**	93.1	**94.1**	**31.6**	95.4

Table 3.2: Precision and recall for r-span and gold mention detection. MD is the span-based single-task mention detector. Gold numbers show that the current models all overgenerate candidate spans; the numbers are also comparable to Table 3.1.

the oracle spans, following the "w/ oracle mentions" model of Lee et al. (2017). All other model parameters follow those from Lee et al. (2018).

Entity mention detection

Table 3.2 reports the precision and recall of the span predictions of the model after the initial span-ranking step on both r-spans and gold coreference spans. While the r-span F1s can be further tuned based on the number of desired candidate spans,[7] for fair comparison, I generate the same number[8] for each model. Additionally, a model trained only for mention detection is included for comparison. These results show that the baseline model is insufficient in capturing noun-like mentions, while a multitask one performs much closer to one trained on just noun-like mentions.

[7] This did not help for coreference resolution.
[8] 0.4 spans per token in the document.

Coreference Resolution

Armed with an improved mention detector, I can now evaluate on the full coreference resolution task.[9] For the pipelined setup, each model uses the spans predicted using the previous models: predicted baseline spans, predicted r-spans, gold r-spans, and gold coref spans. These spans are used in lieu of the candidates generated by each of the models (when both mention generation and linking models are the same, it is equivalent to the end-to-end setup). The results in Table 3.3 primarily highlight the usefulness of gold annotations and the strength of the baseline.

Note that the baseline model is robust to which mentions it receives. While it is possible to improve it by providing gold or gold-derived mentions, the predicted spans are not helpful. Similarly, the mention scorer of the baseline model is also apparently strong, outperforming the end-to-end multitask model and even the gold r-spans in the linking-only model. At the same time, note that the linking-only model is highly reliant on having gold mentions, as even the gold-derived r-spans yield a precipitous drop, especially in precision.

Also, these results suggest that optimizing for r-span F1 may not always lead to performance gains in the pipelined setting, as the models with worse r-span F1 can perform competitively in the coreference task. However, given perfect r-spans, the state of the art can be improved (74.2 F1). With predicted r-spans, this study yields a pipelined model for which the r-span F1 is high and coreference F1 is still near the

[9]The mentions predicted by just the mention detector yielded consistently lower performance, and so are omitted from the table.

Model	MUC			B^3			$CEAF_{\phi_4}$			
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Avg.
c2f (+ l)	82.0	79.2	80.6	72.8	68.7	70.7	68.7	66.7	67.7	73.0
c2f + r	82.4	78.8	80.5	73.2	68.0	70.5	69.4	65.6	67.4	72.8
c2f + r_g	83.6	80.0	81.8	74.6	69.6	72.0	71.1	66.9	68.9	74.2
c2f + g	92.2	81.4	86.5	82.8	71.5	76.7	85.6	63.2	72.7	78.6
Multitask (+ r)	82.0	78.8	80.3	72.4	68.1	70.2	68.9	64.7	66.7	72.4
Multitask + l	81.6	79.1	80.3	71.9	68.7	70.3	68.7	65.2	66.9	72.5
Multitask + r_g	82.9	79.5	81.2	73.0	69.1	71.0	70.2	65.0	67.5	73.2
Multitask + g	91.7	81.0	86.0	81.4	71.0	75.9	85.2	61.6	71.5	77.8
Linking (+ g)	93.5	92.1	92.8	85.8	84.6	85.2	88.3	80.9	84.5	87.5
Linking + l	41.3	77.0	53.7	33.7	66.7	44.8	27.7	67.9	39.4	46.0
Linking + r	29.6	73.8	42.3	22.4	62.8	33.1	20.8	59.0	30.8	35.4
Linking + r_g	36.3	80.8	50.1	28.0	71.1	40.1	26.1	66.8	37.5	42.6

Table 3.3: Performance in a pipeline and multitask setting on the OntoNotes 5.0 test set. Pipelined models (+) are given spans as input. l spans are from Lee et al. (2018); r spans are from Multitask; r_g are the gold r-spans derived from gold annotations; and g are the gold coreference spans. A linking-only model does not train a mention scorer for the first beam step. Nothing is bolded because most rows are not comparable as some models use gold spans.

original baseline.

3.1.5 Discussion and weaknesses

This study attempted to move the spotlight back onto spans for coreference resolution by showing that existing models are insufficient at predicting noun-like mentions that are desired by downstream systems. A mention detector is trained with a multitask objective and used in a pipelined coreference model; these variants are compared to the existing state-of-the-art. While they do not improve on the coreference resolution task itself, the method and model does make significant gains in

detecting the noun-like mentions while still being comparable at coreference resolution.

Besides the original motivation (that singletons are useful), this study also highlights, perhaps, the need for more adoption of datasets that do have singletons annotated. In particular, there is likely a mismatch between the motivation behind r-spans and the types of mentions present in OntoNotes, and that this incompatibility means that improving r-span detection would not affect coreference linking. However, there are now smaller, preexisting datasets with singletons, like PreCo (Chen et al., 2018). Thus, this study could have seen more success with datasets that do contain mentions more similar to r-spans. Indeed, this is what we later find in Toshniwal et al. (2021), when training with the silver "pseudo"-singletons derived from an OntoNotes mention detector (as opposed to the syntactically derived r-spans). Further, the "multitask" training objective used in this study has since been used as a standard method for modeling datasets with singletons both at training and inference (Xu and Choi, 2020; Toshniwal et al., 2020a; Xia and Van Durme, 2021; Yuan et al., 2021).

3.2 A coreference pipeline in information extraction in practice

Note

This section describes a real-world application for coreference resolution. This is work directly related to the DARPA AIDA program and tangentially related to the DARPA KAIROS program. There are contributions towards two systems used for the evaluations. One is towards a pipeline approach described in the previous section, although it was retrained for non-English languages. The other contribution was a (still pipeline) more optimized, multilingual approach to coreference in real-world systems that was subsequently featured in a EACL 2021 Demo paper (Xia et al., 2021) and described more in Section 5.1.

3.2.1 Streaming Multimedia Knowledge Base Population (SM-KBP)

In the SM-KBP (2018, 2019, 2020) task,[10] there are three tracks, or subtasks. In-document knowledge extraction is the focus of the first task, where systems must extract knowledge elements from a stream of English, Russian, and Ukrainian

[10] https://tac.nist.gov/2018/SM-KBP/index.html, https://tac.nist.gov/2019/SM-KBP/index.html, https://tac.nist.gov/2020/KBP/SM-KBP/index.html

documents. For each document, the goal is to produce a knowledge graph containing the entities, events, relations, and their arguments. Naturally, one component of this task is to perform coreference resolution so that the text spans are clustered into (typed) entities that the events and relations can reference for their argument roles. Furthermore, given the disparate tasks and lack of datasets with complete annotations following the specific ontology for this task, training an end-to-end model is infeasible, and a multi-module pipeline system was the engineering approach that we took.

Our first attempt (Xia et al., 2018) was a pipeline where coreference resolution was run independently of the entity typing or the event and relation extraction. In this system, the E2E (Lee et al., 2017) model is used on the raw text (or its translation). Then, a fuzzy aligner merges predictions made by this model and those from the entity typing and event/relation extraction models. Each entity cluster is assigned an entity type based on the majority prediction, and these clusters are subsequently treated as canonical entities participating in the events/relations and as the knowledge elements in the document-level knowledge graph. Even in this setting, some documents (over 5,000 tokens) cannot be processed due to memory constraints and need to be (arbitrarily) split. Even though this accounts for fewer than 5% of the total number of documents, it still called for specific, separate engineering and troubleshooting to handle these rare cases. Furthermore, note that this coreference model does not handle singletons. For this system, this was not detrimental, since unlinked mentions found by the argument extraction models can be upgraded to their own entity element

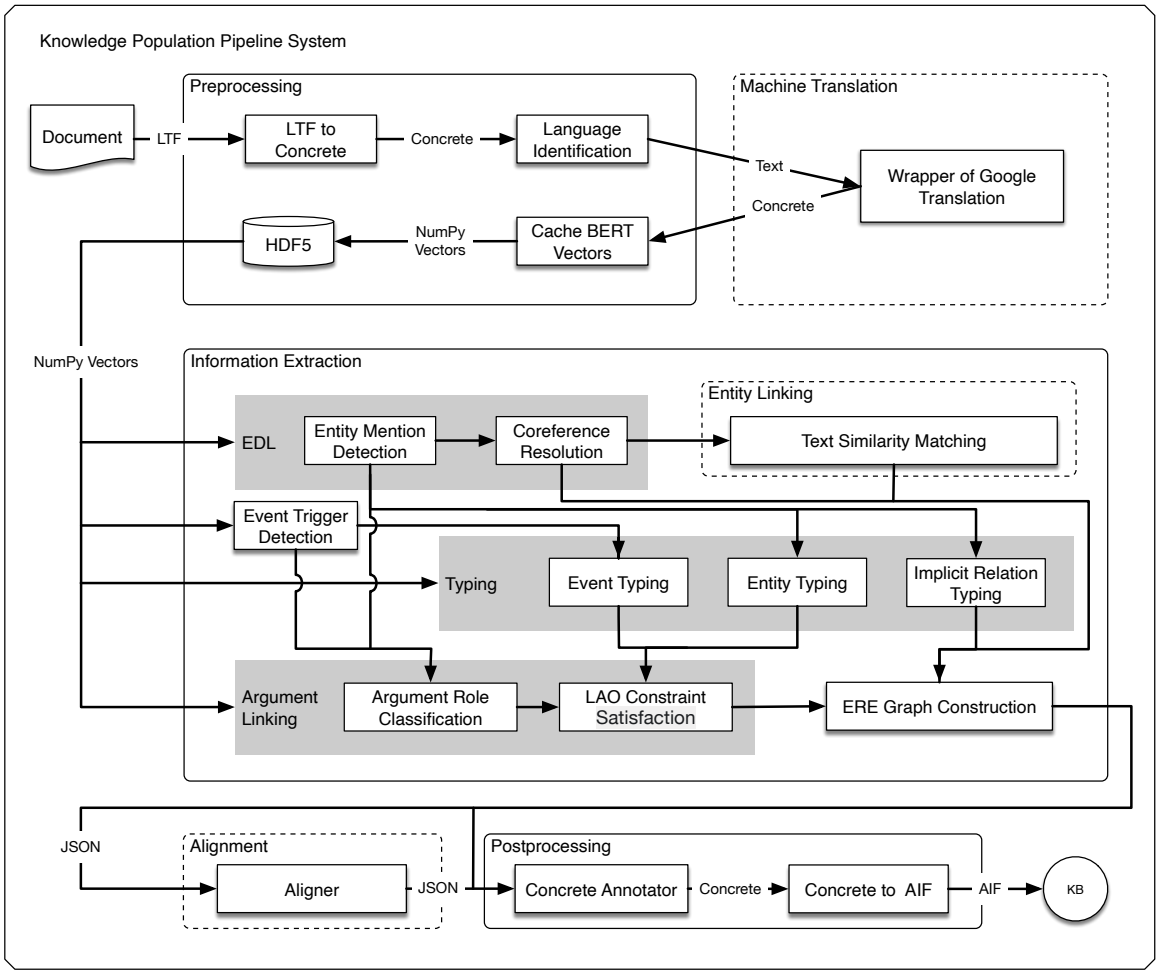


Figure 3.2: This figure, reproduced from Chen et al. (2019), shows the overall system for our team's second submission to the SM-KBP task. Note that coreference resolution (including Entity Mention Detection and Coreference Resolution) plays a tiny role and is neither the first nor last model. Due to its intermediary role, the coreference model may be optimised more for recall or precision. In this case, higher recall of detected entity mentions is desired, if possible, because subsequent systems can filter unused mentions.

in the knowledge graph.

In our second pipeline attempt (Chen et al., 2019), coreference resolution is still an intermediate component of the full system, although it is additionally depended upon for mention detection. Based on the findings of Section 3.1, we used the "*c2f +*

r" approach as it leads to better F1 for mention detection at a relatively small cost on the coreference metrics (-0.2 F1), as it was believed that r-spans were closer to the types of spans in the corpus than the intermediate predictions from the end-to-end model ("*c2f + l*"). This detects mentions first, which can be used for subsequent entity typing models.

The SM-KBP document stream consists of documents in multiple languages. With the rise of multilingual pretrained encoders like mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020), a new approach was viable. Rather than forcing a translation layer somewhere in the pipeline, we can instead train (the same) model multilingually and rely on the strength of cross-lingual transfer. Nonetheless, the remainder of the system (entity typing, argument linking, coreference) was still dependent on separate models, and so a pipeline system is still necessary.

Our last attempt featured a multilingual coreference resolution model that uses XLM-R, a pretrained multilingual encoder (Conneau et al., 2020).[11] However, this was focused on only the *linking* task, using initial spans and entity type predictions from Lin et al. (2020). For this task, evaluation was not based on the typical coreference resolution metrics. Instead, it was based on *typed* entity F1. We found that by re-clustering (without retraining) and using a naive majority vote, we were able to substantially improve scores over the baseline, which is an English-only system (although the details are not available). In English, we improved from 40.39 to 42.81

[11]More details are in Section 5.1.2.

(+2.42); in Russian, we improved from 26.2 to 33.25 (+7.05 F1), and in Spanish, we improved from 21.73 to 24.79 (+3.06). Note that our model did have access to some Russian and Spanish data (see Section 2.2.1 for details). This last attempt resulted in the creation and release of LOME, an information extraction system for which multilingual coreference resolution is one component (Xia et al., 2021) and will be discussed in more detail in Section 5.1.

3.2.2 Discussion

These real-world systems and evaluations demonstrate that coreference resolution models can be useful as both an end-to-end task where predicted mentions are used by other systems and as a linker-only model, which takes in mention boundaries as inputs. Which "mode" to operate in depends on the task, data, and the relative strength of the other parts of the pipeline. For example, the second attempt described in Section 3.2.1 relies on mentions predicted by the coreference model because we believed the linguistic motivation behind r-spans was close to the actual markables in the dataset. For the last attempt (and in LOME), the upstream models were selected for mention prediction as in-domain data was more readily available in that form and mention detection could be more readily learned.

Thus, contrary to the message of Lee et al. (2017) and Section 3.1, treating coreference resolution linking as a sub-task within a pipeline, as opposed to a fully self-contained end-to-end system allows for flexibility in designing information

extraction systems. In other words, there are benefits to the modularity gained by disentangling mention detection and linking, like gaining the ability to interface well with mentions *from* upstream systems or reusing the mentions *in* downstream systems. Second, basic cross-lingual transfer and multilingual coreference models can be created cheaply thanks to cross-lingual abilities of multilingual pretrained models like XLM-R, but they still do not achieve high performance, and especially not in the full coreference setting. This is investigated further in Chapter 5, with the goal of improving cross-lingual transfer. Meanwhile, the theme of using a single model for multiple languages or datasets is revisited more thoroughly in Section 5.1.2 for multilinguality and Chapter 6 in the context of model size and parameter efficiency.

3.3 Takeaways

This chapter retells use cases and raises research questions for coreference resolution beyond chasing the next state-of-the-art model on OntoNotes. In particular, it motivates several questions:

1. If the goal is to better *model* coreference resolution, how can mention prediction (for singletons) be used as an intermediate task? Or, is it really an "artificial" and unrelated task?

2. How can we address the practical issues surrounding inference time and memory? Specifically, coreference resolution models should "just work" on any input, and

therefore have the ability to process long documents without raising memory issues.

3. How do we go beyond OntoNotes to datasets that do annotate singletons? A singletons-friendly model would be a useful component towards building models for pipelined systems.

4. More generally, how can we create coreference resolution systems for *new* datasets or languages? Coreference annotation is expensive.

5. Going further, then, what does a fast, memory-efficient model for arbitrary coreference resolution datasets and languages look like? Recall that, as discussed in Section 2.2, different datasets (even for the same language) disagree on annotation guidelines.

The common thread between these questions is one of efficiency. (1), (2), and (5) can be answered by improvements to the underlying modeling algorithms, which are explored in Chapter 4. (3) and (4) are related to *data* and annotation efficiency, for which continued training is shown to be an effective strategy in Chapter 5. Finally, Chapter 6 also explores (4) and (5) by looking at efficient multitask, or multi-dataset, models.

The methods in this thesis are aimed at improving real-world applications for coreference resolution models. For several years, the progress in the field has remained fairly stagnant as it continues to benchmark on linguistically correct datasets like

English OntoNotes[12] instead of ones directly used by downstream applications. While neural models climb higher on OntoNotes, users of coreference resolution systems still find large performance gaps between the publicly released models and their own needs. One of the non-technical goals and contributions of this thesis is to shift more attention in the sub-field towards general solutions for real-world use cases and problems, and Section 7.2 reflects on the success of that effort.

[12]There's immense intrinsic value here too, as many model advancements *do* transfer over to new domains and languages; however, there is also overfitting to OntoNotes and many of the top models have accrued some research debt for real-world use cases.

Chapter 4

Efficient inference of coreference resolution models

This chapter details ICoref, a constant-memory incremental coreference resolution model. Section 4.2 describes how the model should be extended to predict singleton clusters, while Section 4.3 studies the model in the online setting.[1]

4.1 Constant memory coreference resolution

Note

This work is adapted from "**Incremental Neural Coreference Resolution in Constant Memory**", presented at EMNLP 2020, with João Sedoc and Benjamin Van Durme.

Abstract

This study investigates modeling coreference resolution under a fixed memory constraint by extending an incremental clustering algorithm to utilize contextualized encoders and neural components. Given a new sentence, this end-to-end algorithm proposes and scores each mention span against explicit entity representations created from the earlier document context (if any). These spans are then used to update the

[1]The code for the ICoref model, which is used throughout this thesis, is available at https://github.com/pitrack/incremental-coref/.

entity's representations before being forgotten; the model only retain a fixed set of salient *entities* throughout the document. A high-performing model (Joshi et al., 2020) is successfully converted into a constant-memory version, asymptotically reducing its memory usage to constant space with only a 0.3% relative loss in F1 on OntoNotes 5.0.

4.1.1 Background

Recall that models for coreference resolution typically encode the entire text before scoring and subsequently clustering candidate mention spans which could be either found by a parser (Clark and Manning, 2016b) or learned jointly (Lee et al., 2017). Prior work has primarily focused on improving pairwise span scoring functions (Raghunathan et al., 2010; Clark and Manning, 2016a; Wu et al., 2020) and methods for decoding into globally consistent clusters (Wiseman et al., 2016; Lee et al., 2018; Kantor and Globerson, 2019; Xu and Choi, 2020). Recent models have also benefited from pretrained encoders used to create high-dimensional input text (and span) representations, and improvements in contextualized encoders appear to translate directly to coreference resolution (Lee et al., 2018; Joshi et al., 2019; Joshi et al., 2020).

These models typically rely on simultaneous access to all spans – $\Theta(n)$ for a document with length n – for *scoring* and all scores – up to $\Theta(n^2)$ – for *decoding*. As the dimensionality of contextualized encoders, and therefore the size of span

representations, increases, this becomes computationally intractable for long documents or under limited memory. Given these constraints, expensive scoring functions are increasingly difficult to explore. Further, prior models depart from how humans incrementally read and reason about coreferent mentions; Webster and Curran (2014) argue in favor of a limited memory constraint as a more psycholinguistically plausible approach to reading and model coreference resolution via shift-reduce parsing.

Motivated by scalability and armed with advances in neural architectures, I revisit that intuition by creating a constant-memory coreference model. Following prior work as described in Section 2.5, this model begins with a SpanBERT encoding of a text segment to form a list of proposed mention spans (Joshi et al., 2019; Joshi et al., 2020). Clustering is performed online: each span either attaches to an existing cluster or begins a new one. Memory usage is substantially minimized during inference by storing only the embeddings of the active entities in the document and a small set of candidate mention spans. The two contributions of online clustering and storing of a constant size set of active entities result in an end-to-end trainable model that uses $O(1)$ space with respect to document length while sacrificing little in performance (see Figure 4.1).

4.1.2 Model

The algorithm revisits the approach taken by Webster and Curran (2014) for incrementally making coreference resolution decisions (online clustering). The major

differences lie in explicit entity representations, neural components, and learning.

Baseline

The baseline is derived from the coreference resolution model described by Joshi et al. (2019) with SpanBERT-large (Joshi et al., 2020), which is described in detail in Section 2.5. For each document, the model enumerates all spans, embeds, and scores them. These spans are then ranked and pruned to the top $\Theta(n)$ mentions. For each remaining span, the model learns a distribution over its possible antecedents (via a pairwise scorer) and the training objective maximizes the probability of its gold labeled antecedents. The entire model (including finetuning the encoder) is trained end-to-end over OntoNotes 5.0.

Inference

The proposed method in this study (Algorithm 1) stores a permanent list of entities (clusters), each with its own representation. For a given sentence or segment, the model proposes a candidate set of spans. For each span, a *scorer* scores the *span* representation against all the *cluster* representations. This is used to determine to which (if any) of the pre-existing clusters the current span should be added. Upon inclusion of the span in the cluster, the cluster's representation is subsequently updated via a (learned) function. Periodically, the model evicts less salient entities, writing

them to disk. Under this algorithm, each clustering decision is permanent.[2]

Algorithm 1 FindClusters(Document)

```
Create an empty Entity List, E
for segment ∈ Document do
    M ← Spans(segment)
    for m ∈ M do
        scores ← PairScore(m, E)
        top_score ← max(scores)
        top_e ← argmax(scores)
        if top_score > 0 then
            Update(top_e, m)
        else
            Add_new_entity(E, m)
    Evict(E)
return E
```

Concretely, the incremental coreference (ICoref) model uses a contextualized encoder, SpanBERT (Joshi et al., 2020), to encode an entire segment. Given a segment, Spans returns candidate spans, a result of enumerating all spans up to a fixed width, encoding spans as a combination of the embeddings within the span, and pruning using a learned scorer, following prior work (Lee et al., 2017; Joshi et al., 2019).

PairScore is a feedforward scorer which takes as input the concatenation of a mention span and entity representation along with additional embeddings for distance and genre. Update updates the entity representation ($\mathbf{e}_{top\_e}$) with the newly linked span representation ($\mathbf{e}_m$). This study uses a learned weight, $\alpha = \sigma(\text{FF}([\mathbf{e}_{top\_e}, \mathbf{e}_m]))$ and updates $\mathbf{e}_{top\_e} \leftarrow \alpha \mathbf{e}_{top\_e} + (1 - \alpha)\mathbf{e}_m$.[3] Here, FF is a feedforward network and σ is the sigmoid function.

[2]This uses greedy decoding; exploring decoding strategies is beyond the scope of this work, which is focused on memory. Section 4.3 revisits greedily-made decisions.

[3]Using a simple moving average performs slightly worse.

To ensure constant space, EVICT moves some entities from E to CPU. These entities are never revisited; the offsets are stored on CPU solely for evaluation purposes. The evictions are based on cluster size and distance from the end of the segment.

The algorithm is independent of these components, so long as they satisfy the correct interface. Specifically, this algorithm is compatible with entirely different models, like the one by Wu et al. (2020). They use a query-based pairwise scorer, which could be adopted in place of the feedforward pairwise scorer. The use of abstract components in this algorithm also allows for comparison of different encoders or update rules.

Training

Similar to prior work (Lee et al., 2017), the training objective is to maximize the probability of the correct antecedent (cluster) for each mention span. However, rather than considering *all* correct antecedents, the objective is only interested in the cluster for the *most recent* one. Scoring is between mention spans and entity clusters, so there should be a single correct cluster.[4] For each mention m, *scores* is treated as an unnormalized probability distribution $P(e \mid m)$ for $e \in E$, where E is the entity list that includes an ε target label which represents the action of starting a new cluster. The exact objective is to maximize $P(e = e_{\text{gold}} \mid m)$; e_{gold} is the gold cluster of m (i.e., the cluster the most recent antecedent was assigned to).[5]

[4]This assumption is revisited later, in Section 4.2. Maximizing for all antecedents works well too.
[5]See Section 4.2 for a correction that maximizes *all* antecedents instead.

However, the entirely sequential algorithm also introduces sample inefficiency, as most mentions have the same label (ε) and do not accrue significant loss. Training is sped up by accumulating gradients periodically, trading computation time for space. This tradeoff is similar to that of batching by documents, which is impractical from a memory perspective. Like prior work, the parameters are updated once per document (and not once per mention).[6]

For comparability, pretrained components are prioritized: not only are the encoder weights that are already finetuned on this dataset reused as initialization for the encoder, but also the mention and pairwise scorers from Joshi et al. (2020) are used as initialization for Spans and PairScore. The implementation of Joshi et al. (2020) and Joshi et al. (2019) was the most amenable to extension and experimentation and therefore serves as the illustrative example of converting a memory-intensive model into a constant memory one.

4.1.3 Experiments

Since the weights from Joshi et al. (2020) (the baseline) are reused, the main experiment is to compare their model to the constant space adaptation in both task performance and memory usage. Additionally, I analyze document and segment length, conversational genre, and explicit clusters.

[6]It can also be updated once per segment, especially if the corpus contains few, but long, documents.

Data

This study focuses on OntoNotes 5.0 (Weischedel et al., 2013; Pradhan et al., 2013), which consists of 2,802, 343, and 348 documents in the training, development and test splits respectively. These documents span several genres, including those with multiple speakers (broadcast and telephone conversations) and those without (broadcast news, newswire, magazines, weblogs, and the Bible). Later sections and chapters of this thesis will reuse this model for other datasets.

Implementation

The model dimensions and training hyperparameters match the baseline model, a publicly available coreference resolution model by Joshi et al. (2019) and Joshi et al. (2020). As discussed, their (trained) parameters are used as initialization for the encoder, span scorer, and span pair scorer. However, ICoref does not make use of speaker features, since it is not meaningful to assign a speaker to the cluster representation. At the end of each segment, the model evicts singleton (size 1) clusters more than 600 tokens away from the end of the segment. Additionally, it evicts all clusters whose most recent member is more than 1200 tokens away. In this work, the encoder is frozen—further finetuning the encoder provided little, if any, benefit likely because the encoder has already been finetuned on this dataset and task. This is in contrast to subsequent work, both in the field and in this thesis, which will find benefits via finetuning. Additional details, including our choice of eviction function,

are described in Section 4.1.8. All experiments are performed on either a single NVIDIA 1080 TI (11GB) or GTX Titan X (12GB).

4.1.4 Results

4.1.4.1 Performance

Table 4.1 presents the OntoNotes 5.0 test set scores for the metrics: MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and $CEAF_{\phi_4}$ (Luo, 2005) using the official CoNLL-2012 scorer. I reevaluated the baseline and report the scores for CorefQA directly from Wu et al. (2020). These is a small drop in performance compared to the baseline and no apparent drop with eviction.

	MUC			B^3			$CEAF_{\phi_4}$			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
Joshi et al. (2020)	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	79.6
Ours	85.7	84.8	85.3	78.1	77.5	77.8	76.3	74.1	75.2	79.4
Ours (without eviction)	85.7	84.9	85.3	78.1	77.5	77.8	76.2	74.2	75.2	79.4
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 4.1: Complete results of our model on the OntoNotes 5.0 test set with three coreference resolution metrics: MUC, B^3, and $CEAF_{\phi_4}$. For completeness, scores for the contemporaneous (and still current) state-of-the-art are included. All models use an encoder derived from SpanBERT-large.

4.1.4.2 Document Length

The goal of this study is a constant-memory model that is comparable to the baseline. Table 4.2 shows the average F1 broken down based on the length (in

Subset	#Docs	**JS-L**	Ours	Δ	-evict
All	343	80.1	79.5	-0.6	79.7
0-128	57	84.6	84.5	-0.1	84.5
129-256	73	83.7	83.6	-0.1	83.6
257-512	78	82.9	83.4	+0.5	83.4
513-768	71	80.1	79.3	-0.8	79.3
769-1152	52	79.1	78.6	-0.5	79.0
1153+	12	71.3	69.6	-1.7	69.8
1 Speaker	268	81.1	81.0	-0.1	81.2
2+ Speakers	75	76.7	75.0	-1.7	75.0
Test	348	79.6	79.4	-0.2	79.4

Table 4.2: Average F1 score on the development set broken down by document length and number of speakers. **JS-L** refers to the `spanbert_large` model from Joshi et al. (2020), which is the baseline, and -evict refers to the model without eviction.

subtokens)[7] of the document and number of speakers. ICOREF is competitive on most document sizes and in the single speaker setting. On longer documents, eviction has a minor effect. Because ICOREF does not make use of speaker embeddings, it performs worse on documents with multiple speakers. This drop due to speaker features matches previous findings (Lee et al., 2017). One way to include speakers and retain speaker-independent entity embeddings is by treating speakers as part of the input text (Wu et al., 2020) and adopted in Section 4.3.

4.1.5 Inference Memory

Next, we can look towards space. Table 4.3 shows the GPU memory needed to perform inference over the entire development set. Compared to the baseline and its

[7]This split of the development set differs from that used by Joshi et al. (2019) which counts the number of 128-subtoken sized segments. I directly count subtokens.

Model	GPU Memory (GB)	Dev. F1
ICoref (1 sent.)	1.7	66.7
ICoref (10 sent.)	2.0	77.1
ICoref (128 toks.)	1.6	74.7
ICoref (512 toks.)	2.0	79.5
No eviction	2.0	79.7
JS-B	6.4	77.7
JS-L	>11.9	80.1

Table 4.3: Space needed and performance over the development set. **JS-B** and **JS-L** refer to the `base` and `large` variants SpanBERT used in the baseline.

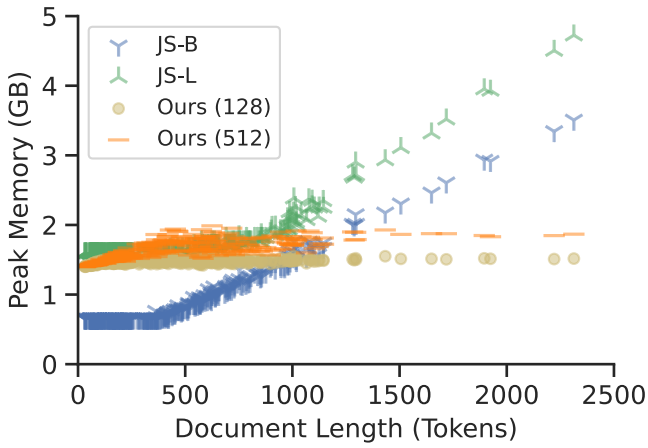


Figure 4.1: Total size of GPU-allocated tensors for each document in the development set. The base (**JS-B**) and large (**JS-L**) models of the baseline use apparently linear space, while ours with inference segment lengths of 128 and 512 use constant space.

smaller `base` version, ICoref uses substantially less memory. In practice, eviction also has little effect on memory and F1 on this dataset. However, it is a necessary part of guaranteeing that the model will not crash on arbitrarily long inputs, which is objective 2.) from Section 3.3.

Usage in practice is subject to the memory allocator, and this implementation (PyTorch) differs in framework from the baseline (TensorFlow). To fairly compare

the two models, I computed the maximum *space used by the allocated tensors* for each document during inference.[8] Figure 4.1 compares this value of peak theoretical memory usage of several models against the dataset. It shows the baseline is dominated by a term that grows linearly with length, while that is not the case for our model, which has constant space usage.

ICoref reduces the asymptotic memory usage to $O(1)$. In addition, these plots do not clearly show asymptotic memory usage: the baseline and other derivative models have a quadratic component for scoring span pairs (with a small coefficient). The encoder, SpanBERT, adds a significant constant term (with respect to document length) to all models. While there is some work in sparsifying Transformers (Child et al., 2019; Kitaev et al., 2020), there (still) does not yet exist a sparse SpanBERT, which would be useful in this setting. Chapter 6 discusses some efforts in pruning (rather than sparsifying) the encoder to reduce the inference time and memory.

These plots show that models have relatively modest memory usage during inference. However, their usage grows in training, due to gradients and optimizer parameters. This additional memory usage would render training and finetuning the underlying encoder infeasible for the baseline but possible using ICoref with 12GB GPUs.

[8]For profiling, I use `run_op_benchmark` for TensorFlow 1.15 and `pytorch_memlab` 0.0.4 and `torch.cuda` for PyTorch 1.5.

4.1.6 Segment Length

The memory usage at each step (and therefore of the algorithm) is also dependent on the segment length due to the encoder. Table 4.4 explores the effect of the length of each segment (split at sentence boundaries), which gives further insight into the tradeoff between performance and memory reduction. These models are compared without eviction and the confirm observations from Joshi et al. (2019) that larger context windows compatible with the encoder input size improve performance. Additionally, models trained on shorter sequences can be scaled, at inference time, to longer sequences and obtain gains in performance. There is an unsurprising substantial drop using single sentences, owing to coreference being largely a cross-sentence phenomenon.

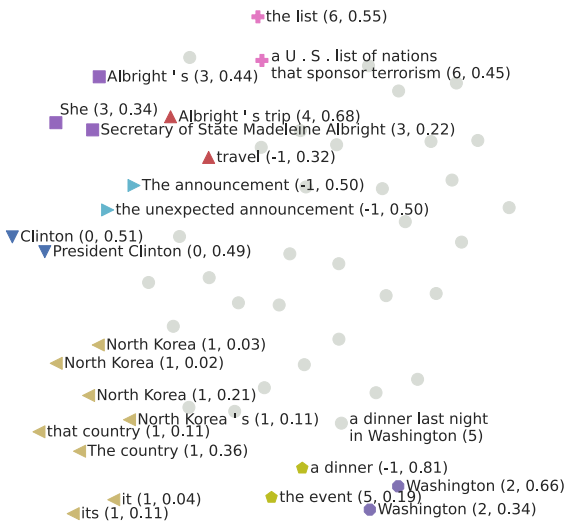
		Inference Length			
	Train↓	1 sent.	10 sent.	128 toks.	512 toks.
sents.	1	70.0	76.4	75.2	76.9
	10	68.9	77.8	76.2	78.9
toks.	128	70.1	77.2	76.3	77.7
	256	69.1	77.9	76.5	78.8
	384	67.7	77.3	76.1	79.1
	512	67.1	77.7	75.6	79.7

Table 4.4: Average dev. F1 score for models trained and evaluated across a range of segment lengths (either fixed number of sentences or subtokens).

4.1.7 Span Representations

Figure 4.2 visualizes the proposed span representations for a single document in the development set. The colors/shapes represent our predictions, and each point is

annotated with the text, the gold cluster label, and the (normalized) α for each span (recall α is used in the UPDATE function to determine a span's contribution to its entity embedding).



President Clinton may travel to North Korea in an attempt to improve relations with that country. The announcement comes after two days of talks between American and North Korean leaders in Washington. Secretary of State Madeleine Albright has accepted an invitation to visit North Korea and meet with leader Kim Jong-il. She made the unexpected announcement at a dinner last night in Washington. North Korea's top defense official hosted the event. The country is on a U.S. list of nations that sponsor terrorism. The Clinton administration is trying to persuade North Korea to halt its ballistic missile program as a way it can get off the list. There's no word yet when Albright's trip will take place.

Figure 4.2: t-SNE plot (left) of span representations of a single document (right) in the development set (cnn_0040_0). Each color/shape is a predicted cluster, while light gray circles indicate predicted singletons. For each span, the gold cluster label (-1, if not annotated) and its contribution to the entity embedding is noted in parentheses.

Given these embeddings, the figure supports the viability of clustering approaches: gold coreference clusters tend to be "close" in embedding space. Regarding α, some spans are weighted equally ("Clinton") while others are not ("North Korea"). This could be a result of online updates biasing more recent spans with higher weights. Alternatively, it may suggest that some spans (like names) are more informative than others (like pronouns).

4.1.8 Technical Appendix

This section describes implementation details and additional experiments. Gradients are accumulated until memory usage exceeds 7.5GB. One drawback of ICoref is that, because of its sequential nature, it is considerably (up to 2x) slower than pairwise-scoring models, like most variations of e2e. In initial trials, I explored sampling losses for negative examples (spans that do not have an antecedent). While sampling with a rate of 0.2 (for example) would speed up training and inference, ultimately it contributed up to a one point deficit in F1.

I also explored teacher forcing, in which spans are added to the gold cluster during training instead of the predicted one. This would "correct" the training objective to match prior work. However, this did not have a noticeable effect on performance in this setting. Likewise, it is also possible to train a competitive model for which only the SpanBERT encoder from Joshi et al. (2019) was retained and the span scorer and pairwise scorer were randomly initialized. However, this was not chosen for the final experiments because training in this setting is more time-intensive. Further, learning span detection is not guaranteed by this objective, leading to high variance across runs (most notably in the number of epochs). Thus, the effect of other hyperparameters would not be as apparent.

Additionally, I attempted further finetuning the encoder with a separate learning rate of [1e-5, 5e-6], but was unsuccessful in improving the performance. Training (without finetuning) roughly takes 70 min/epoch with negative sample rate 0.2, 100

min/epoch without sampling loss, and 160 min/epoch when finetuning. All runs are stopped after 5 to 15 epochs due to early stopping (patience = 5).

For eviction, a policy which evicts singletons distance > 600 and all clusters distance > 1200 would have a recall of 99.57% over the training set. This is a result of sweeping over [200, 300, 400, 500, 600, 900] for singletons and [400, 600, 800, 1000, 1200, 1800] for all clusters. I also tried using a single fixed distance, as well as other non-constant schemes (e.g. size × distance as thresholds). Here, distance is between the current point in the document and the average of the start and end indices of the most recent span added to the cluster. I selected this policy from several other choices due to the recall it achieved.

The ICoref model dimensions otherwise match up exactly with Joshi et al. (2019). Rather than omitting the speaker embedding and segment length embedding entirely (which would affect pairwise scorer dimensionality), these embeddings are replaced with **0**.

Concretely, I performed grid searches over dropout ([0.3, 0.4, 0.5]), sample rate ([0.2, 0.5, 0.75, 1.0]), and update method ([alpha, mean]). I find that 0.4 dropout, 1.0 sample rate, and alpha weighting were the best after 2 epochs. Alpha weighting resulted in, on average, approximately 0.1 F1 improvement (after 2 epochs).

Alpha weighting uses a two-layer MLP: the first layer has size 300 and ReLu nonlinearity, while the final layer then projected to a scalar with a sigmoid activation. After fixing those values, I explored learning rate ([5e-5, 1e-4, 2e-4, 5e-4]), eviction

policy at training ([no eviction, eviction]), and gradient clipping value ([1, 5, 10]). The best combination is 2e-4, no eviction, and gradient clipping at 10, which performs slightly better, although there was little difference between them after these models were allowed to converge. Given the final set of hyperparameters, I performed five training runs, resulting in average development set F1 of [79.4, 79.5, 79.5, 79.5, 79.7]. the best-performing model for the results in this study. For Table 4.4, each model was only trained once.

For these experiments, ICoref contains 377M parameters, of which 340M is SpanBERT-large (Joshi et al., 2020).

4.1.9 Conclusions

This study proposes an online algorithm for space efficient coreference resolution that incorporates contributions from recent neural end-to-end models. It demonstrates how to transform a model which performs document-level inference into an incremental algorithm. Doing so greatly reduces the memory usage of the model during inference at virtually no cost to performance, thereby providing an option for researchers and practitioners interested in modern coreference resolution models for tasks constrained by memory, like the modeling of book-length texts.

Additionally, the inference (and training) algorithm implicitly creates singleton clusters for discourse-new referents. This naturally decomposes the coreference resolution model into an intermediate mention detection task which is *directly* used

in the model instead of merely as an auxiliary objective. Nonetheless, it is not hard to modify ICoref to properly predict singletons, described in in the next section, Section 4.2

4.2 Adapting ICoref beyond OntoNotes

This section is a short formal description of the ICoref model and how it can be extended to predict singletons at inference, along with some mathematical justification for the training objective.

Model formalization

Given a text segment of length n with (sub)tokens $x_1 \dots x_n$, ICoref enumerates all spans $x_{a,b} \in X$, where $x_{a,b} = [x_a, x_{a+1}, ..., x_b]$ up to a certain length, respecting sentence boundaries. The span embedding $\mathbf{x}_{a:b}$ is then computed as a function of the component embeddings, determined by the output of an encoder: $\mathbf{x}_{a:b} = [\mathbf{x}_a; \mathbf{x}_b; f([\mathbf{x}_a, ..., \mathbf{x}_b]); \phi(a, b))]$ where f is an attention-weighted average and $\phi(a, b)$ is a width feature. This is identical to the representation used by Lee et al. (2017). Like prior work, $s_m(x_i)$ is a learned scoring function intended to rank the likelihood the given span is a coreference mention.

ICoref iterates through the spans, collecting a list of clusters, C (initially empty). Each span x_i is scored by a pairwise scorer, $s_c(x_i, c)$, against the clusters already found by the model. Specifically, $s_c(x_i, c) = s_m(x_i) + s_a(x_i, c)$, which means this score is

influenced by the likelihood x_i is a mention. This is akin to the pairwise antecedent scorer from prior work. However, in ICoref, the scores are computed against clusters instead of against spans, which reduces the need for cluster decoding later.

If $\max_{c_j \in C}(s_c(x_i, c_j)) \leq 0$, a new cluster, $c_{\text{new}} = \{x_i\}$ with embedding $\mathbf{x}_i$, is created and added to C. Otherwise, x_i is merged into the top-scoring c_j, with the new embedding,

$$\mathbf{c}'_j = \alpha \mathbf{x}_i + (1 - \alpha)\mathbf{c}_j,$$

where α is a learned function of x_i and c_j.

The training objective aims to minimize $-\log \prod_{x_i \in X} P(c^*_{x_i}|x_i)$, where $c^*_{x_i}$ is the correct cluster determined by the cluster containing the most recent antecedent of x_i. If no such antecedent exists, then the correct cluster is the dummy cluster, ϵ, and $s_c(x_i, \epsilon) = 0$. Letting $C_\epsilon = C \cup \{\epsilon\}$, the probability can then be computed as

$$P(c^*_{x_i}|x_i) = \frac{\exp(s_c(x_i, c^*_{x_i}))}{\sum_{c_j \in C_\epsilon} \exp(s_c(x_i, c_j))}.$$

Training objective correction

The training objective can be modified to optimize for *all* antecedents of x, $Ant(x)$, instead of the most recent one:

$$-\log \prod_{x_i \in X} \sum_{y_i \in Ant(x_i)} \frac{P(c_{y_i}|x_i)}{|Ant(x_i)|}. \tag{4.1}$$

This leads to comparable (or slightly better) performance than using the most recent antecedent. Note that this is not quite the same, mathematically, as the objective in the E2E model which also optimizes for all antecedents. In that objective, the antecedent probabilities are all weighted equally, while here, clusters containing more correct antecedents are weighted higher.

Genres

Finally, s_a usually incorporates a genre embedding determined by the genre of the document (within OntoNotes). We still retain that small set of parameters for compatibility reasons but assume all documents have the same genre. This genre embedding is rarely used (only for OntoNotes downloaded models), and future work can be focused on better incorporation of parameters specific to a single genre, domain, or task – especially when looking towards multi-purpose models (e.g. multilingual models in Section 5.1.2 or multitask models in Chapter 6).

Singletons

For most datasets and many downstream tasks, singleton entity mentions need to be predicted. For OntoNotes, all singleton mentions would be removed in postprocessing.[9] Prior work adds an auxiliary objective that maximizes $s_m(x_i)$ if x_i is an entity mention (Section 3.1, Zhang et al. (2018b)) and only prune out singleton mentions $s_m(x_i) < 0$ in postprocessing. Instead, the reformulation presented next is similar to the choices

[9] Or, more dangerously, ignored in the evaluation script.

made by Toshniwal et al. (2020b) and possible due to incremental nature of the model.

Instead of taking the top kn of all spans at span pruning, the algorithm prunes to the top kn spans only from the set $\{x_i \in X : s_m(x_i) > 0\}$ (which could have fewer than kn elements). In other words, it only retains spans that are predicted to be mentions and is faster during the forward pass. Now, the training objective is to minimize $s_m(x_i)$ if x_i is not an entity mention, and maximize $s_m(x_i) + s_a(x_i, c^*_{x_i})$ if it is, where $c^*_{x_i}$ is the gold cluster. This latter term is identical to $s_c(x_i)$ from the earlier formulation.

Mathematically, this change can be interpreted as now modeling the joint distribution of whether x_i is an entity mention (a binary random variable M) and which entity cluster (E) is the best match, according to s_a. The joint probability decomposes to,

$$P(E, M \mid x_i) = \sum_{m \in \{0,1\}} P(E|m, x_i) P(m, x_i).$$

This can further split into the components,

$$P(E|M = 1, x_i) = \frac{\exp(s_a(x_i, c^*_{x_i}))}{\sum_{c_j \in C_\epsilon} \exp(s_a(x_i, c_j))} \tag{4.2}$$

$$P(E|M = 0, x_i) = 1 \tag{4.3}$$

$$P(M = 1, x_i) = \frac{\exp(s_m(x_i))}{1 + \exp s_m(x_i)} \tag{4.4}$$

$$P(M = 0, x_i) = 1 - P(M = 1, x_i) \tag{4.5}$$

The $M = 1$ objective is the same as training without singleton mentions (as in OntoNotes), while the $M = 0$ term accounts for singletons. Note that if $M = 0$, then we can always make the correct "cluster" decision by ignoring it for the remainder of the algorithm, which allows for this simplification.

This is different from simply adding an objective maximizing $P(M)$, since that would incorrectly handle cases when $M = 0$. In practice, however, this does not affect accuracy on the task, although it validates the approach of pruning spans earlier, which results in a faster model.

4.3 Low-latency online coreference resolution

Note

This work is adapted from "**Online Neural Coreference Resolution with Rollback**," which was presented at CRAC 2022, with Benjamin Van Durme. It is a modeling extension to ICoref, as it resolves the drop in performance that arises for short segment sizes and is tested in the streaming, online setting, which is common in conversational contexts. This results in a model that is operable fully *online* with only a small drop in F1.

Abstract

Humans process natural language online, whether reading a document or participating in multiparty dialogue. Recent advances in neural coreference resolution have focused on offline approaches that assume the full communication history as input. This is neither realistic nor sufficient to support dialogue understanding in real-time. This study benchmarks two existing, offline, models and highlights their shortcomings in the online setting. These models are modified to perform online inference and *rollback*, a short-term mechanism to correct mistakes, is introduced. The effectiveness of this approach is validated across five datasets against an offline and a naive online model in terms of latency, final document-level coreference F1, and average running F1.

4.3.1 Motivation and Introduction

In environments like multiparty spoken dialogue and social media streams, text in the form of tokens and sentences are available in (near) real-time. To promptly make use of this data, NLP systems often need to process text before additional tokens or sentences are available. For example, this could enable interruptions with a response or a clarification question (Boyle et al., 1994; Li et al., 2017), make decisions during a social media stream (Mathioudakis and Koudas, 2010), or recognize and translate speech live (Oda et al., 2014; Ma et al., 2020). While some language technologies

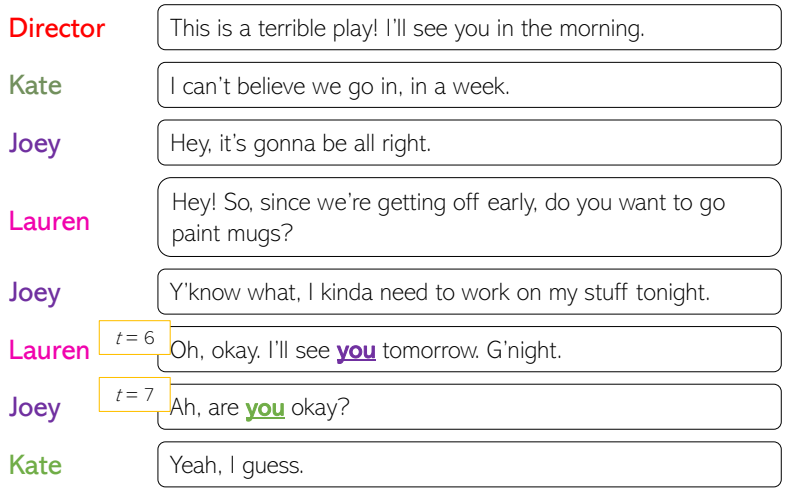


Figure 4.3: In this scene from *Friends*, viewers can deduce who "you" refers to at $t = 6$, and we want coreference models to be similarly capable. At $t = 7$, viewers may need more context, such as the identity of the next speaker, to be certain of who "you" refers to. Absent that context for a text-based model, its predictions will be incorrect. *Rollback* is a cheap and local revision mechanism that corrects these type of mistakes.

operate incrementally in the *online* setting, many document-level understanding models and tasks do not.

A core task in language understanding is resolving references. Recent work has made significant progress on improving accuracy for single documents (Lee et al., 2017; Wu et al., 2020) and in the cross-document setting (Caciularu et al., 2021). However, this focus on document-level resolution makes use of global higher order inference and document-level encodings. As interest in coreference resolution is shifting back towards dialogue (Khosla et al., 2021), the *offline* setting is inconsistent with how dialogue is found in the real world. Now equipped with neural models and large-scale data, I revisit the *online* coref setting (Stoness et al., 2004; Schlangen et al., 2009).[10]

[10]Contemporaneously, Xu and Choi (2022) study the online setting.

This work is motivated by the human ability to resolve references *without looking into the future* (Figure 4.3). The online scenario is simulated for two offline models (Xu and Choi, 2020; Xia et al., 2020a) by making full predictions after each sentence and masking the future context. This either leads to significantly increased latency or lowered accuracy. The latter model (ICoref) is then modified to properly perform online inference and while accuracy does drop relative to the offline baselines, the latency is substantially lower. To ameliorate this, *rollback* is introduced, which is a backtracking method that allows the model to correct recently made decisions. On several coreference datasets, this can recover performance comparable to that of the offline model with the latency of online models.

4.3.2 Task: Online Coreference Resolution

In offline (single doc) coref, the input is a document D, and the output is a set of clusters (or chains) of text mentions, $\mathcal{C} = \{C_1, ..., C_n\}$ such that any two mentions in a given C_i corefer. Document-level evaluation, $S(\mathcal{C}_{\text{pred}}, \mathcal{C}_{\text{gold}})$, compares the predicted clusters to the gold reference clusters with an average three corpus-level metrics (MUC, B^3, and CEAF_{ϕ_4}) for the accuracy of mentions, links, and clusters. When each metric is instead computed at the corpus level instead before averaging, we refer to this as *final F1* (identical to CoNLL 2012 F1).

In the sentence-level online setting, $D = [X_1, X_2, ..., X_T]$ is a stream of sentences or utterances. After time t, the model needs to predict clusters $\mathcal{C}_{\text{pred},t} = \{C_{1,t}, ..., C_{n,t}\}$

conditioned on only $[X_1, ...X_t]$. The reference clusters, $\mathcal{C}_{\text{gold},t}$, are restricted to contain only mentions up to sentence X_t. This may lead to empty clusters which are ignored when calculating the score.[11] To evaluate accuracy in the online setting, I propose additionally computing a *running F1* for each document which is averaged across all time t,

$$S_{\text{running}}(\mathcal{C}_{\text{pred}}, \mathcal{C}_{\text{gold}}) = \sum_{t=1}^{T} \frac{1}{T} S(\mathcal{C}_{\text{pred},t}, \mathcal{C}_{\text{gold},t}).$$

These document-level scores are subsequently averaged across the corpus (macro-average), in contrast to the already corpus-level metrics of *final F1*.

This is not the first study to observe that references should be resolvable without future context. Prior work (Stoness et al., 2004; Schlangen et al., 2009; Poesio and Rieser, 2011) has also emphasized the importance of incremental (online) prediction of reference, especially in the context of dialogue. Since most models at that time already operated at the sentence level, their work is at the token-level granularity. This study does not go as far; the goal is to first rein back *document* level neural models to the sentence level, which is still appropriate in applications where full utterances are available.

Another measure of performance is the latency of different systems. Unlike token-level work in speech (Zhang et al., 2016) or translation (Gu et al., 2017), this study is primarily interested sentences, especially as timestamps are not readily

[11] Singletons may also be ignored depending on the convention in the dataset.

available. Furthermore, modern models can process a single sentence in under a second, while sentences take substantially longer to be spoken or typed. Therefore, I chose to primarily report document-level latency, which is the *wait time* between the end of the document and production of predictions. This choice is revisited and discussed sentence-level latency in Section 4.3.4.4.

4.3.3 Method

4.3.3.1 Datasets

As one goal is to analyze a variety of domains, several coreference datasets are studied.[12] The CoNLL 2012 Shared Task (OntoNotes) (Pradhan et al., 2013) is split into into the conversational (telephone and broadcast conversations) and nonconversational text (newswire, newsgroups, broadcast news, weblogs, religious texts) genres. Character Identification (CI) (Zhou and Choi, 2018) consists of transcripts from the TV show *Friends* and is another source of social and informal conversations. LitBank (Bamman et al., 2020) is a collection of long excerpts from literature, which allows us to study latency scaling. Finally, QBCoref (Guha et al., 2015) is a collection of trivia questions where players are expected to interrupt with the answer, which is an example of a task needing a fast NLU model.

[12]These were introduced in more detail in Section 2.2.1.

Preprocessing

The preprocessing follows that of prior work: Joshi et al. (2019) for OntoNotes, Xia and Van Durme (2021) for LitBank (first fold) and QBCoref (first fold), and Toshniwal et al. (2021) for CI. For the genre split in OntoNotes, while the full dataset into a conversational and text-based component, some weblog documents are also conversations on message boards. The genre-based split is justified because the weblog documents less conversational than spoken dialogue. While OntoNotes does have non-English splits, only English data is used in this study.

Since ICoref does not readily take speaker embeddings, the underlying text of CI is augmented with speakers by prepending each utterance with the name of the speaker(s), following the strategy outlined by Wu et al. (2020), and these prefixes are filtered out before evaluation. There could be other ways of representing the speakers, especially in plural situations, which is beyond the scope of the work. While this follows the same preprocessing as Toshniwal et al. (2021), this does not need to be done for the c2f model because speakers are used directly as a feature. CI evaluation uses the conventional CoNLL 2012 score instead of the one outlined in Zhou and Choi (2018) because the goal is to explore online coreference and high-level trends by using the dialogue and conversational nature of the dataset and not focus on the plural mentions and multiparty aspect.

4.3.3.2 Models

C2F (implemented by Xu and Choi (2020)) and ICOREF (Section 4.1) are used as the offline baselines. The inference procedure of the latter is modified for the online experiments.

C2F (Xu and Choi, 2020) is a reimplementation of the coarse-to-fine coreference model (Lee et al., 2018) which detects mention spans in the entire document, scores them with each other, and finds the most likely antecedent for each span. It then uses higher order decoding strategies to promote pairwise consistency within a cluster. This study does not use these higher order decoding strategies because they are slower and only improve performance slightly. I do, however, use the extension of the training loss that accommodates singletons (Xu and Choi, 2021).

ICoref (Xia et al. (2020a), Section 4.1) is a memory-efficient incremental coreference resolution model, itself a variant of the C2F model. The model naturally segments the document into pieces and incrementally processes each piece. After each text segment, the predictions for that segment are committed. This hard decision foregoes any higher-order decoding strategies, but this locality offered is exactly what should be extended in the sentence-level online setting.

Naive online C2F is a baseline where C2F is used to make full predictions after every sentence. For a document with n sentences, this costs n calls to the full C2F model, and effectively acts as an upper limit on model performance.

Online ICoref. For the online models, the inference process in ICOREF is modified.

Like prior models, ICoref encodes a variable number of sentences per encoder forward pass, and each sentence would have access to future contexts. The fully online version of the algorithm segments the text by sentences instead of by tokens. Thus, instead of making predictions every fixed number of tokens, predictions are written every u sentences. Setting $u = 1$ would make an online model at the sentence level.

Online ICoref with rollback. A drawback of both ICoref and online modeling in general is the inability to correct mistakes in light of future context. This study proposes "rollback," which is run every r sentences (Algorithm 2). This process reverts all predictions made in the previous r sentence-window and remakes them all, batch-mode, with the full (r-sentence) context. The trade-off of increasing r is that the intermediate prediction quality can suffer, while decreasing r incurs additional latency.

4.3.4 Experiments and Results

First, I show that current models rely on future context, which is not readily available in the online setting. Next, I demonstrate the effectiveness of online models under latency and average running F1. In particular, I analyze the benefits of rollback. Finally, I verify that for reasonable input stream speeds, online approaches are indeed appropriate.

4.3.4.1 Masking the future

I first investigate the dependence of the two baseline (offline) models, C2F and ICoref, on future context. As shown in Figure 4.3, models often use future contexts to make predictions such as linking "you" with the next speaker. For each model, a sentence-level causal mask is applied to the encoder and global or future-looking decoding algorithms are removed. The causal mask restricts each token's attention only to other tokens in its sentence or a previous one. With this mask at inference, performance of both models drops considerably (Table 4.5; full version in Table 4.6). However, by finetuning with the causal mask, the C2F model recovers from these drops in the masked setting. This suggests that coreference resolution models can be retrained to make better use of previous context and rely less on "easy" future signals. This finding is also quite promising for future investigation into *training* methods.

On the other hand, masked training does not appear to affect the performance of the ICoref model. Nonetheless, the incremental nature of ICoref is more amenable to extension to an online setting, and ICoref (without masking) is used as the model to be adapted into the online setting.

4.3.4.2 Online inference strategies

To properly evaluate online performance (as opposed to only simulating masking the future), the modified ICoref (as described in Section 4.3.3.2) is evaluated on running F1, final F1, and wait time. I find that increasing update sizes, u, interpolates

Δ Final F1	C2F		ICoref	
Masked Training?	No	Yes	No	Yes
OntoNotes^{conv}	-7.8	-1.8	-8.0	-7.6
OntoNotes^{text}	-6.0	-0.3	-8.0	-6.9
LitBank	-5.3	-1.9	-5.1	-5.4
QBCoref	-4.9	-0.5	-1.1	-2.7
CI	-5.5	-1.0	-11.0	-9.6

Table 4.5: A model is trained with and without sentence-level causal attention masks. This table reports the difference in F1 between inference with and without these mask in the offline setting. The numerical results are also reported in Table 4.6.

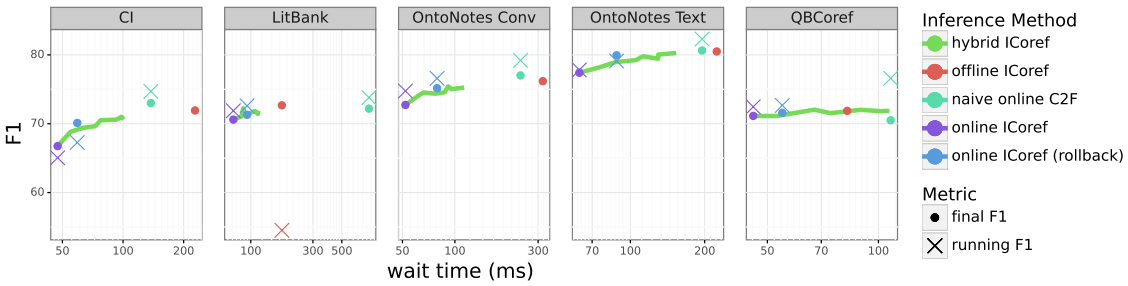


Figure 4.4: These plots show the average wait time against the final F1 (test) and the running F1 (×) for select models. Varying the update frequency **interpolates** between **online** and **offline** ICoref models in both final F1 and wait time. The **naive online C2F** baseline is also included for comparison. The proposed method of **rollback** offers a strong compromise with higher F1s and comparable wait times vs. the fastest online models, and a final F1 comparable to the offline models.

Algorithm 2 Online coreference resolution with rollback

Input: Sentences $S = s_1, s_2, ...$; update frequency u; rollback frequency r; initial clusters $\mathcal{C}_0 = \emptyset$.
for $s_t \in S$ **do**
  if $t \equiv 0 \pmod{ur}$ **then**
    $\mathcal{C}_{t-ur+1} = \textsc{Revert}(\mathcal{C}_{t-1})$
    $\mathcal{C}_t = \textsc{ICoref}(S[t-ur+1:t], \mathcal{C}_{t-ur+1})$
  else if $t \equiv 0 \pmod{u}$ **then**
    $\mathcal{C}_t = \textsc{ICoref}(S[t-u+1:t], \mathcal{C}_{t-1})$
  else
    $\mathcal{C}_t = \mathcal{C}_{t-1}$
  yield $\mathcal{C}_t$

Δ Final F1	C2F				ICoref			
Masked Training?	No		Yes		No		Yes	
Masked Inference?	Yes	No	Yes	No	Yes	No	Yes	No
OntoNotes^{conv}	69.2	77.0	75.0	76.7	68.2	76.2	68.4	76.0
OntoNotes^{text}	74.7	80.6	79.9	80.2	72.5	80.5	73.4	80.3
LitBank	66.9	72.2	68.8	70.7	67.6	72.7	67.5	72.9
QBCoref	64.9	69.8	70.0	70.5	70.8	71.9	69.7	72.5
CI	67.6	73.0	71.8	72.8	60.9	71.9	61.2	70.9

Table 4.6: This is the full version of Table 4.5, on the test set. Each entry instead shows the score with mask and the score without mask instead of the difference.

between an online model ($u = 1$) and the unmasked offline ICoref model (where u is the encoder window size). This "hybrid" mode trades off wait time for F1, as increasing u leads to longer wait times but better performance. Additionally, changing the rollback frequency does not correlate with time because larger updates are both costlier and rarer. So, the best r is chosen based on each dev set.

Figure 4.4 shows that the online and hybrid models are faster than the offline ICoref model and do better on running F1, but worse on final F1. Online with rollback is usually the best approach, as it achieves high F1 scores across all datasets,

Metric	naive online C2F			ICoref		
	Run. F1	Fin. F1	wt (ms)	Run. F1	Fin. F1	wt (ms)
OntoNotes^{conv}	79.2	77.0	237.8	24.9	76.2	319.3
OntoNotes^{text}	82.3	80.6	195.2	28.9	80.5	223.8
LitBank	73.8	72.2	807.4	54.5	72.7	173.3
QBCoref	76.6	70.5	107.9	15.6	71.9	82.3
CI	74.7	73.0	137.5	14.2	71.9	227.8
Metric	**Online ICoref**			**+ rollback**		
	Run. F1	Fin. F1	wt (ms)	Run. F1	Fin. F1	wt (ms)
OntoNotes^{conv}	74.8	72.7	52.0	76.6	75.2	79.0
OntoNotes^{text}	77.8	77.4	62.1	79.1	79.9	87.9
LitBank	71.9	70.6	73.5	72.6	71.3	93.7
QBCoref	72.5	71.1	45.8	72.7	71.6	54.9
CI	65.1	66.7	47.3	67.3	70.1	59.3

Table 4.7: Final F1, running F1, and wait time for each datasets and four inference algorithms. The proposed rollback mechanism offers a strong compromise with higher F1s and comparable wait times vs. the fastest online models, and a final F1 comparable to offline ICoref. Naive online C2F is the strongest method, but also the slowest.

while it also has short wait times. Naive online C2F performs well on F1, but it is substantially slower on especially short or long documents.

The small margin on QBCoref could be explained by the fact that the forward pass for online ICoref is equal to that of a causally masked offline model and Table 4.5 shows that the gap between a masked and unmasked model is small.

Dataset	#Edits	Ment.	New	Existing
LitBank	453	12.1, 9.3	12.6, 10.4	27.2, 6.0
QBCoref	145	20.0, 8.3	16.6, 13.1	16.6, 7.6
CI	429	4.9, 4.4	17.0, 5.6	27.0, 13.3

Table 4.8: The edits made in each dev set via rollback are categorized: **Ment**ion detection errors, missed **New** clusters, and incorrect links to **Existing** clusters. We report the percentage of (wrong→right, right→wrong) edits. The unreported fraction of edits are wrong→wrong.

4.3.4.3 Error correction with rollback

In Table 4.8,[13] I record the number of predictions that are changed with rollback.[14] In general, more edits are corrections (wrong→right) than errors (right→wrong), which demonstrates the effectiveness of rollback. For all three datasets, many of the corrections made address correctly assigning spans to existing clusters, such as the "you" in Figure 4.3. In QBCoref, many corrections are un-predicting a non-mention, while in CI, many corrections are correctly predicting new starts of entity clusters.

4.3.4.4 Latency analysis

The running assumption in this work is that each sentence arrives after all computation has been completed for the previous sentence, which motivates wait time as a metric. However, this assumption may not always be true in situations where utterances are highly frequent or short, like in online chat rooms.

[13]OntoNotes is omittted because they remove singleton clusters before evaluation, making this type of analysis difficult.

[14]Each mention identified by the model either before or after rollback is split based on its gold reference antecedent: not a mention, discourse-new, or part of another cluster. For the first two classes, this counts the number of revisions. For the third, a cluster link is correct if the majority of the predicted cluster overlaps with the reference cluster.

To verify this empirically, I run simulations to find the token arrival rate for which offline and online models have equivalent *sentence* latency. To compute sentence-level latency, (sub)token arrivals are assumed to be uniform at a fixed prespecified rate. When the last token of a sentence arrives, if the model decides to process the preceeding chunk, I simulate running inference over the previous sentence(s). In parallel (in this simulation), tokens continue arriving.

The latency metric of interest is the time between the end of *each sentence* and when the predictions *for that sentence* are produced by the simulated model. Since ICoref is sequential, if the model is due to process a segment before the previous one is completed, the next segment is blocked until the previous one is complete. To simulate this, the forward pass is run once to obtain the size of the job for each of these segments, and then used to simulate sentence-level latency with different token arrival rates. The goal is to gain some intuition over token arrival rates. To demonstrate the extremes and collect a lower bound on the arrival speed of tokens for which offline models will have less latency, the online and offline ICoref models, which were usually the fastest and slowest were the targets of the simulation (Figure 4.5).

For all datasets, the point at which offline and online models have equivalent sentence latency is at over 200 words per second (wps). Additionally, if the stream is slower than 20 wps, there is never a "delay" caused by processing a sentence. This is substantially faster than the speaking (Yuan et al., 2006) and reading (Brysbaert, 2019) rates of around 3-5 wps. Therefore, sentence-level predictions are being made

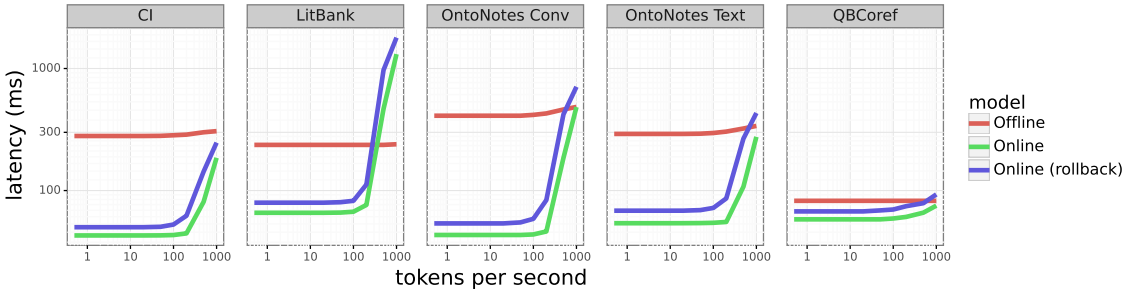


Figure 4.5: Simulated mean sentence-level latency given different token arrival rates.

faster than tokens are produced, which validates our metric of wait time in this work. This may not extend to some settings with high arrival rates, like livestream comments.

4.3.5 Technical Appendix

The default hyperparameters are used for both the C2F model and ICoref. For C2F, I tried training with mention detection loss (coefficient=1), as it may help with singletons. It has a small effect in training, and so for QBCoref and LitBank, we use a mention detection loss. In addition, following the previous findings on continued training (Gururangan et al., 2020; Xia and Van Durme, 2021), each model continues training from the publicly released OntoNotes checkpoints. Each model is trained once, as the goal is to highlight online coreference resolution, specifically, online *inference*.

To that end, I also explore several values of $u \in [1, 2, 3, 4, 5, 6, 7, 8]$ and $r \in [2, 4, 5, 6, 8, \text{no rollback}]$ for each of the datasets. The results for various values of u is reported in Figure 4.4, which is an interpolation between the online and offline models. The best values were: $r = 4$ for QBCoref, $r = 6$ for LitBank, and $r = 8$ for

the other splits. Furthermore, following the findings in Section 4.3.4.1, all models are trained with and without the causal mask, although in practice models without the mask performs better.

For each test set and model (i.e. plotted point in Figure 4.4), inference is run three times and the *minimum* time is reported rather than the average. Minimum is used because in rare cases, one of the runs would be significantly slower, which would disproportionately affect the average. Overall, the mean difference between the max and min wait time across all datasets is around 10.5ms, or 12% relative to the min wait time, and the median is 5.8ms.

All experiments are run on a single NVIDIA RTX Quadro 6000 GPU. Training each model completes in under 24 hours, with some datasets like QBCoref taking significantly less times (under an hour). Inference takes 1-5 minutes per trial.

4.3.6 Conclusions and Limitations

Section 4.3 looks at reining back document-level models for neural coreference resolution to the utterance level by proposing a shift towards online inference. I propose a model with the capability for making predictions online, after every sentence. This leads to lower latency than a corresponding offline model, and maintains a consistently high running F1 after each sentence. To edit predictions made without future context, I introduce a rollback mechanism which reverts and corrects recently made predictions, bringing the F1 closer to that of the offline model while maintaining

its ability to make online predictions with low latency.

This is still only the first steps towards efficient inference for coreference resolution, especially as the underlying encoders grow increasingly large (and slow). Future steps would consider extensions to this approach by handling online processing at the word-level, revisiting the scenario considered by Schlangen et al. (2009). Furthermore, it would be exciting to connect this to a real-world application and demonstrate benefits in efficiency for downstream tasks like semantic parsing or dialogue understanding.

Chapter 5

Improving data efficiency via model transfer

This chapter contains two studies, Section 5.1 uses the efficient model from Chapter 4 as part of a full information extraction system. This leads to investigating (and finding deficiencies) around multilingual models. Section 5.2 looks closer at methods for transferring models across domains and languages for the full coreference resolution task.

5.1 A multilingual information extraction system

Note

LOME: Large Ontology Multilingual Extraction was a system built by a large team at Johns Hopkins University and University of Rochester and demonstrated at EACL 2021 System Demos.[1] Information on using the Docker container, web demo, and demo video at `https://nlp.jhu.edu/demos`. I contributed to parts of the full system demonstration, and I have additional multilingual findings not featured in the original paper that are discussed in Section 5.1.2.

[1]Guanghui Qin contributed equally by creating the FrameNet model and standing up the demo site.

5.1.1 An overview of Large Ontology Multilingual Extraction (LOME)

LOME (Xia et al., 2021) is a system for performing multilingual information extraction. Given a text document as input, our core system identifies spans of textual entity and event mentions with a FrameNet (Baker et al., 1998) parser. It subsequently performs coreference resolution, fine-grained entity typing, and temporal relation prediction between events. By doing so, the system constructs an event and entity focused knowledge graph. We can further apply third-party modules for other types of annotation, like relation extraction. Our (multilingual) first-party modules either outperform or are competitive with the (monolingual) state-of-the-art. We achieve this through the use of multilingual encoders like *XLM-R* (Conneau et al., 2020) and leveraging multilingual training data. LOME is available as a Docker container on Docker Hub. In addition, a lightweight version of the system is accessible as a web demo.

As information extraction capabilities continue to improve due to advances in modeling, encoders, and data collection, we can now look (back) toward making richer predictions at the document-level, with a large ontology, and across multiple languages. Li et al. (2020a) noted that despite a growth of open-source NLP software in general, there is still a lack of available software for knowledge extraction. We wish to provide a starting point that allows others to build increasingly comprehensive document-level

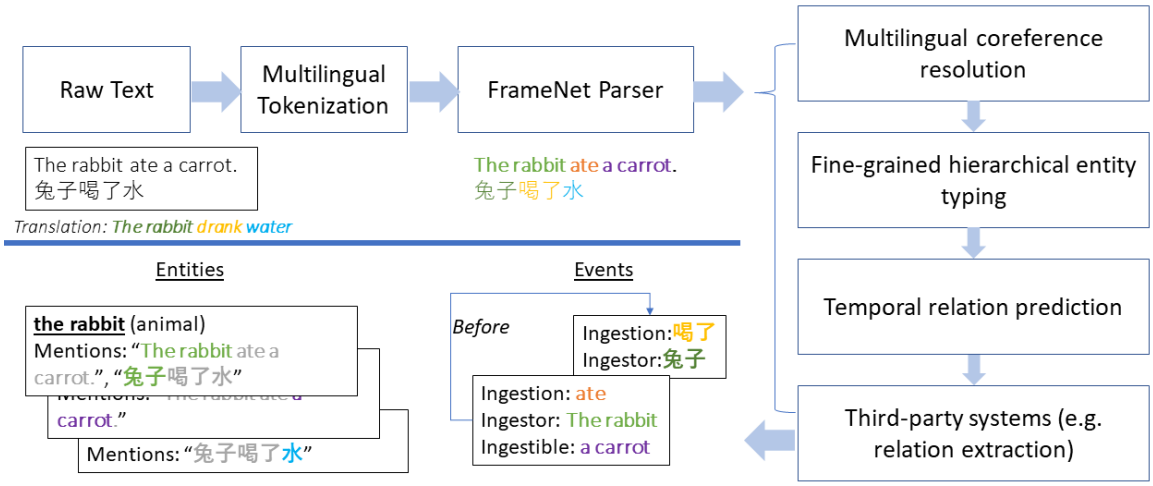


Figure 5.1: Architecture of LOME, reproduced from Xia et al. (2021). The system processes text documents as input and first uses a FrameNet parser to detect entities and events. Then, a suite of models enrich the entities and events with additional predictions. Each individual model can be trained and tuned independently, ensuring modularity of the pipeline. Annotations between models are transferred using CONCRETE, a data schema for NLP.

knowledge graphs of events and entities from text in many languages.

Therefore, we demonstrate LOME, a system for multilingual information extraction with large ontologies. Figure 5.1 shows the high-level pipeline by following a multilingual input example. A sentence-level parser identifies both INGESTION events and their arguments. To connect these events cross-sententially, the system clusters coreferent mentions and predicts the temporal relations between the events. LOME, which supports fine-grained entity types, additionally labels entities like **the rabbit** with LIVING_THING/ANIMAL.

Several prior packages have also used advances in state-of-the-art models to build comprehensive information extraction systems. Li et al. (2019) present an event, relation, and entity extraction and coreference system for three languages: English,

Russian, and Ukrainian. Li et al. (2020a, GAIA) extend that work to support cross-media documents. However, both of these systems consist of language-specific models that operate on monolingual documents after first identifying the language. On the other hand, work prioritizing coverage across tens or hundreds of languages is limited in their scope in extraction (Akbik and Li, 2016; Pan et al., 2017).

Like prior work, LOME is focused on extracting entities and events from raw text documents. However, LOME is language-agnostic; all components prioritize multilinguality. Using *XLM-R* (Conneau et al., 2020) as the underlying encoder paves the way for both training on multilingual data (where it exists) and inference in many languages.[2] Our pipeline includes a full FrameNet parser for events and their arguments, neural coreference resolution, an entity typing model over large ontologies, and temporal resolution between events.

Our system is designed to be modular: each component is trained independently and tuned on task-specific data. To communicate between modules, we use Concrete (Ferraro et al., 2014), a data schema used in other text processing systems (Peng et al., 2015). One advantage of using a standardized data schema is that it enables modularization and extension. Unless there are annotation dependencies, individual modules can be inserted, replaced, merged, or bypassed depending on the application. We discuss two example applications of our Concrete-based modules, one of which further extracts relations and the other performs cross-sentence argument linking for

[2] *XLM-R* itself is trained on CommonCrawl data spanning one hundred languages.

events.

The primary application of LOME is to extract an entity- and event-centric knowledge graph from a textual document. In particular, we are interested in using these graphs to support a multilingual schema learning task (KAIROS[3]) for which data has been annotated by the LDC (Cieri et al., 2020). As a result, some parts of LOME are designed for compatibility with the KAIROS event and entity ontology. Nonetheless, there is significant overlap with publicly available datasets.

Clearly, LOME is much larger in scope than the focus of this thesis. However, this subsection provides context on how (and why) an efficient multilingual model is integral to a fast and functioning system (and demonstration). The next subsection describes some specific findings related to the coreference resolution component.

5.1.2 Multilingual Coreference Resolution

The coreference resolution model is based on ICoref (Section 4.1) because it achieves near state of the art performance with additional benefits. The main motivation for this model choice is robustness: LOME needs the ability to soundly run on all document lengths, and so ICoref is favored over other slightly better performing but brittler systems. In addition, because this coreference resolution model is part of a broader entity-centric system, the module used in this system

[3]This goal is to develop a system that identifies, links, and temporally sequences complex events. More information at https://www.darpa.mil/program/knowledge-directed-artificial-intelligence-reasoning-over-schemas.

does not perform the mention detection step (which is left to the FrameNet parser). Instead, both training and inference assumes given mentions, and the primary task here is mention *linking*. However, there are also some findings for the full multilingual coreference task.

Coreference Linking

ICoref is trained with *XLM-R* (large) as the underlying encoder and with additional multilingual data. Unlike that work, gold spans are provided. This necessitated by the location of coreference in LOME, as mentions are passed to the coreference module as input. In addition, while I previously used a frozen encoder, I find that finetuning improves performance.[4] Finally, The multilingual data used consists of the full OntoNotes 5.0 (Weischedel et al., 2013; Pradhan et al., 2013), a subset of SemEval 2010 Task 1 (Recasens et al., 2010), and two additional sources of Russian data, RuCor (Toldova et al., 2014) and AnCor (Budnikov et al., 2019).

The performance of our model on each language is benchmarked with the average F1 (MUC, B^3, and $CEAF_{\phi_4}$) by language in Table 5.1. The model's performance can also be compared to monolingual gold-mention baselines, where they exist. For English, the gold-mention baseline is an identical model using SpanBERT (Joshi et al., 2020) instead.

That model achieves 92.2 average (dev.) F1, compared to the 92.7 of the

[4]Using AdamW and a learning rate of 5×10^{-6}.

Language	# Training	# Eval Docs	-ru	All
Arabic[O]	359	44	69.4	**70.7**
Catalan[S]	829	142	**66.6**	65.7
Chinese[O]	1810	252	89.5	**90.2**
Dutch[S]	145	23	65.0	**65.1**
English[O]	2802	343	92.4	**92.8**
Italian[S]	80	17	**56.4**	56.0
Spanish[S]	875	140	66.1	**66.5**
Russian[A]	573	127	77.2	**79.2**

Table 5.1: Average F1 scores by language trained with gold mentions, with and without Russian data in training. The superscripts O indicates data from OntoNotes 5.0 (dev), S indicates data from SemEval 2010 Task 1 (dev), and A is the AnCor data (test). Note: this table differs from that of Xia et al. (2021), as that paper reports numbers specifically for the model checkpoint in the demo. For reasons unclear to me now, the model included in the demo performs slightly worse in most languages.

multilingual model. There is also a comparable system for Russian AnCor from Le et al. (2019), which achieves 79.9 F1 using the model from Lee et al. (2018) and RuBERT (Kuratov and Arkhipov, 2019), which is comparable to the that of the multilingual model (79.2). This shows that a single, multilingual model can perform comparably to monolingual models, with the advantage that with a single model, it does not need to perform language ID and is a fraction of the size of a system with one model per language. This finding mirrors prior findings showing multilingual encoders are strong cross-lingually (Wu and Dredze, 2019).

Additionally, the benefits of encoder finetuning can be more carefully investigated by sweeping over layers to determine whether there is benefit to finetuning the entire *XLM-R* large model.[5] This sweep, shown in Figure 5.2, confirms that finetuning more

[5]This question is explored a little differently in Chapter 5.

layers (except the vocabulary embeddings) leads to generally better improvements in score, although a big improvement already arises from finetuning even the top 4 or 8 layers.

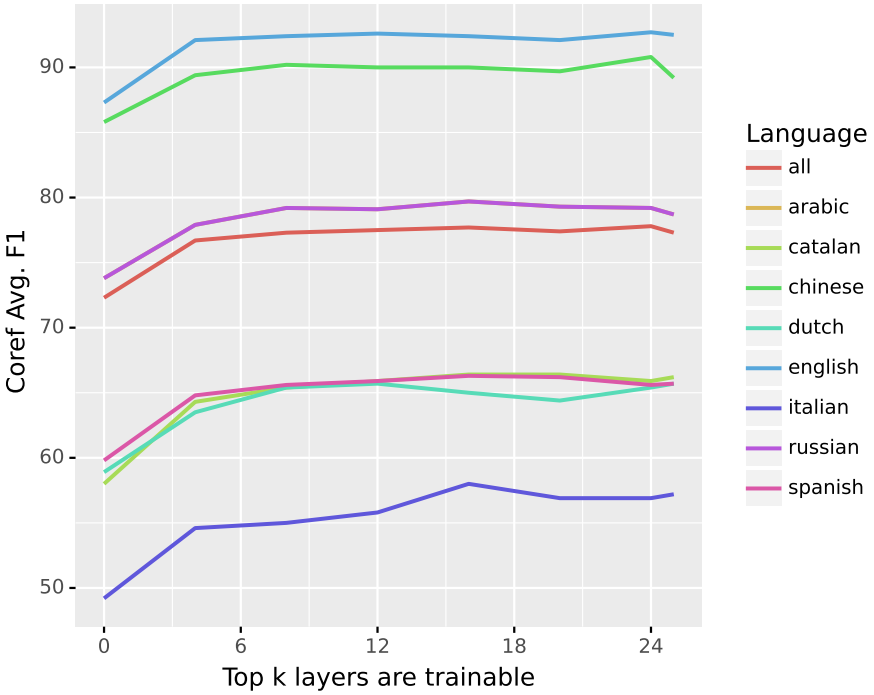


Figure 5.2: Performance of multilingual coreference linking model as the number of trainable layers in *XLM-R* is increased.

Full Coreference Resolution

While not reported in Xia et al. (2021), a similar multilingual model was trained on the full coreference resolution task using the OntoNotes (en, zh, ar) and SemEval 2010 (it, nl, ca, es) datasets. This is the same model as the one in the previous section, except the model is no longer provided gold mentions. One constraint was that because the language is not known at inference it is not possible to pick

language-specific hyperparameters. Notably, this affects the the "maximum span length" considered by the model. I chose 15 (sub)tokens for these experiments, reported in Table 5.2. Unsurprisingly, finetuning the (full) encoder also leads to better results. Disappointingly, in this setting, none of the scores reported are close in performance compared to monolingual models. This motivates the following section Section 5.2, which partially aims to improve scores on each dataset without annotating additional data.

Language	Frozen	Finetuned
Arabic^{O}	23.1	41.4
Catalan^{S}	32.2	55.3
Chinese^{O}	51.7	68.0
Dutch^{S}	32.8	50.9
English^{O}	61.3	77.1
Italian^{S}	19.5	42.4
Spanish^{S}	32.9	57.1

Table 5.2: Average F1 scores by language trained without gold mentions, with and without a finetuned *XLM-R* encoder. The superscripts O indicates data from OntoNotes 5.0 (dev), S indicates data from SemEval 2010 Task 1 (dev).

5.2 Model transfer

Note

This section is adapted from **Moving on from OntoNotes: Coreference Resolution Model Transfer** presented at EMNLP 2021.[6] Code and pretrained

[6]This work was done in 2021 with Benjamin Van Durme.

models for this section are available at https://nlp.jhu.edu/coref-transfer.

Abstract

Academic neural models for coreference resolution are typically trained on a single dataset, OntoNotes, and model improvements are benchmarked on that same dataset. However, real-world applications of coreference resolution depend on the annotation guidelines and the domain of the target dataset, which often differ from those of OntoNotes. I aim to quantify transferability of coreference resolution models based on the number of annotated documents available in the target dataset. I examine eleven target datasets and find that continued training is consistently effective and especially beneficial when there are few target documents. Doing so establishes new benchmarks across several datasets, including state-of-the-art results on PreCo.

5.2.1 Introduction

Starting initially with neurally-learned features (Clark and Manning, 2016a; Clark and Manning, 2016b), end-to-end neural models for coreference resolution (Lee et al., 2017; Lee et al., 2018) have been developed and imbued with the benefits from contextualized language modeling (Joshi et al., 2019; Joshi et al., 2020) and additional pretraining (Wu et al., 2020). At the same time, the number of parameters used in these models have increased, raising questions of overfitting our research to a specific

dataset. Several studies show that fully-trained neural models on preexisting large datasets do not transfer well to new domains (Aktaş et al., 2020; Bamman et al., 2020; Timmapathini et al., 2021), and that rule-based baselines can still be superior (Poot and Cranenburgh, 2020). Further, while prior work has analyzed fully-trained models for mention pairs, like gender bias (Rudinger et al., 2018; Webster et al., 2018; Zhao et al., 2019), there has not been a comprehensive comparison analyzing transfer across datasets for document-level coreference resolution.

This study aims to bridge the current gap in understanding between the strength of pretrained models in contrast to the value of annotated target data, in light of the strong few-shot capabilities demonstrated by pretrained language models (Brown et al., 2020b; Schick and Schütze, 2021). While transfer in other NLP tasks have been studied more in-depth, transfer in coreference resolution has scarcely been examined despite recent models containing hundreds of millions of parameters. I investigate model transfer across datasets with continued training, in which a fully-trained model on a source dataset is further trained on a small number of target dataset examples (Sennrich et al., 2016; Khayrallah et al., 2018).[7]

This work contributes first study of neural coreference resolution transfer, showing that continued training is effective on eleven datasets spanning different domains, annotation guidelines, and languages. I find evidence that OntoNotes, a widely-used but license-requiring dataset for benchmarking coreference resolution, is no better

[7] I use *continued training* to refer to full model adaptation, in contrast to *finetuning* which is more strongly associated with encoders that are trained without supervision (Hinton and Salakhutdinov, 2006).

for model transfer than the freely-available PreCo. Additionally, continued training establishes modern (neural) benchmarks on several understudied datasets, including state-of-the-art results on PreCo. Finally, I analyze practical considerations regarding model selection, catastrophic forgetting, and parameter sharing.

5.2.2 Coreference Resolution

As discussed in Chapter 2, entity coreference resolution is the challenging task of finding clusters of mentions within a document that all refer to the same entity. Specifically, as discussed in Section 2.2.2, annotation guidelines for coreference resolution differ across these datasets based on the intended goals of the creators. Despite such differences, OntoNotes 5.0 (Weischedel et al., 2013) emerged as the most widely-used benchmark for the full task, and widely used public models are based on this dataset (Manning et al., 2014; Gardner et al., 2018). Table 2.2 (from Section 2.2.1) shows some of the concrete differences between OntoNotes and a few other datasets considered in this work.

However, I've argued in this thesis that OntoNotes-based models are not always appropriate. OntoNotes is a collection of several thousand documents across just seven genres from the 2000s (or earlier), and many datasets fall outside of the scope of those genres or time period. And, unlike other datasets, singletons are not annotated. In modeling OntoNotes, genre and speaker features are needed to improve on the state-of-the-art, both of which are idiosyncrasies of the OntoNotes dataset. It is

unclear how well these models transfer to a new, target dataset, especially if it is annotated and usable in (continued) training.

Prior work on domain adaptation for coref has focused on a single dataset and often with non-neural models. Yang et al. (2012) use an adaptive ensemble which adjusts members per document. Meanwhile, Zhao and Ng (2014) use an active learning approach to adapt a feature-based coref model to be on par with one trained from scratch while using far less data. Moosavi and Strube (2018) study model generalization by including carefully selected linguistic features, aiming to improve out-of-the-box general performance. Aktaş et al. (2020) adapt a model to Twitter by retraining with a target-dependent subset of genres of OntoNotes.

While these studies shed insight on single datasets, this chapter aims to set broader expectations and guidelines on effectively using new data for model adaptation, both in terms of quantity and allocation of data between training and model selection.

5.2.3 Methods

5.2.3.1 Continued Training

I adopt the formulation of continued training from Luong and Manning (2015) where a model is first trained on a source dataset until convergence. This fully-trained model is then used to initialize a second model which is trained on a target dataset.

This framework has been used for other tasks where annotation guidelines or

domains shift significantly between datasets, like in syntactic parsing (Joshi et al., 2018), semantic parsing (Fan et al., 2017; Lialin et al., 2021) and neural machine translation (Luong and Manning, 2015; Khayrallah et al., 2018). In addition, continued training can be staggered at different granularities (Gururangan et al., 2020) or use mixed in-domain and out-of-domain data (Xu et al., 2021).

5.2.3.2 Incremental Coreference Model

For this work, it is helpful to view end-to-end models for coreference resolution as having four parts: a text encoder, a scorer for mention *detection*, a scorer for mention pair *linking*, and an algorithm for decoding clusters. The ICoref model (Xia et al., 2020a) used in this work uses SpanBERT (Joshi et al., 2020), followed by mention detection and linking as described in Chapter 4, and decoding is done greedily (offline, without rollback). In particular, the encoder is a pretrained component that is substantially larger (in size) than all the *mention detection and linking* parameters in the other three parts. This model was chosen because of its competitive performance against the line of end-to-end neural coreference resolution models (Joshi et al., 2019) and memory efficiency, which allows for experiments on longer documents.

However, ICoref, like the models before it, is originally designed around OntoNotes. As a result, the minor modifications described in Section 4.2 are made for this study.

Dataset	Training	Dev	Test	# Folds
OntoNotes^{en}	2,802	343	348	-
OntoNotes^{zh}	1,810	252	218	-
OntoNotes^{ar}	359	44	44	-
PreCo	36,120	500	500	-
LitBank	80	10	10	10
QBCoref	240	80	80	5
ARRAU^{RST}	335	18	60	-
SARA	138	28	28	7
Semeval^{ca}	829	142	167	-
Semeval^{es}	875	140	168	-
Semeval^{it}	80	17	46	-
Semeval^{nl}	145	23	72	-

Table 5.3: Number of documents for each of the datasets considered in this work. For the smaller datasets, we perform k-fold cross-validation.

5.2.3.3 Data

This study explores a total of two source datasets and eleven target datasets, described in Table 5.3. For smaller datasets, evaluation is performed via k-fold cross-validation, following the original authors.

OntoNotes 5.0 (Weischedel et al., 2013) is a dataset spanning several genres including telephone conversations, newswire, newsgroups, broadcast news, broadcast conversations, weblogs, and religious text. The dataset contains annotations of syntactic parse trees, named entities, semantic roles, and coreference. Notably, however, it does not annotate for singleton mentions, while it does link events. It also includes data in English (^{en}), Chinese (^{zh}), and Arabic (^{ar}), which is referred to using superscripts.

PreCo (Chen et al., 2018) is a dataset consisting of reading comprehension passages

used in test questions. The authors argue that because its vocabulary is smaller than that of OntoNotes, it is more controllable for studying train-test overlap. While they detail many ways in which their annotation scheme differs from OntoNotes, notably, they annotate singleton mentions and do not annotate events. Furthermore, this corpus is sufficiently large that it is possible to train a general-purpose coreference resolution model. Finally, because the official test set has not been released, the official "dev" set is used as the test set, and a separate 500 training examples as the "dev" set.

LitBank (Bamman et al., 2020) is an annotated dataset of the first, on average, 2,000 words of 100 public-domain books. While they annotate singletons, they also limit their mentions only to those which can be assigned an ACE category.

QBCoref (Guha et al., 2015) is a set of 400 quiz bowl[8] literature questions that are annotated for coreference resolution. This dataset also includes singleton annotations, and it only considers a small set of mention types. The documents are short and dense with (nested) entity mentions, as well as terminology specific to literature questions.

ARRAU (Uryupina et al., 2020) is the second release[9] of ARRAU, a corpus first created by Poesio and Artstein (2008) which spans several genres. The fine-grained annotations mark the explicit type of coreference, and the dataset also includes phenomena like singleton mentions and non-referential mentions. This study only uses the coarsest-grained coreference resolution of the `RST` subcorpus, which is a subset of the Penn Treebank (PTB) newswire documents, and therefore uses the same splits as

[8]Quiz bowl is a trivia competition where passages give increasingly easier hints towards a common answer, such as a book title, author, location, etc.

[9]`LDC2013T22`

PTB (Poesio et al., 2018). Thus, this dataset overlaps with OntoNotes, which also includes sections of PTB. However, ARRAU is used to analyze *annotation* transfer.

SARA v2 (Holzenberger and Van Durme, 2021) is a collection of legal statutes in which text spans identified as arguments of legal structures are also annotated for coreference. Each document is a single short legal statute, and so the overall number of clusters is low while many clusters are singletons.

SemEval 2010 Task 1 (Recasens et al., 2010) is a dataset for multilingual coreference resolution for studying the portability of coref systems across languages. It consists of data in English (overlapping with OntoNotes), German, Spanish (^{es}), Catalan (^{ca}), Italian (^{it}), and Dutch (^{nl}). Due to dataset overlaps and licensing, only the latter four languages are used in this paper.

Dataset Preprocessing

Following prior work (e.g. Joshi et al. (2019)), all documents are processed into sentence-separated and subtokenized segments of sizes at most 512. For all English datasets, the SpanBERT tokenizer is used, while the XLM-R tokenizer is used for the cross-lingual experiments.

For QBCoref, the dataset is split into five splits after shuffling the initial dataset. For LitBank, the published splits are used (Bamman et al., 2020). In ARRAU^{RST}, several mentions are "split" (non-consecutive). Correctly modeling split spans is an active area of ongoing work (Yu et al., 2020a; Yu et al., 2021). Since ARRAU^{RST}

primarily used for intrinsic comparisons, I defer to the *minimum* span if a mention is split. This means I replaced a subset of markables, listed in Table 5.4. In addition, a small number of markables do not have an annotated coreference cluster, while a couple split markables failed to reduce because there is no minimum span annotated. These two phenomena did not affect the test set. Nonetheless, the model's inability to address split markables affects comparability against prior work.

Split	Total	Split	No "coref"	No "min"
train	57,686	677	4	2
dev	3,986	40	0	0
test	10,341	145	0	0

Table 5.4: Statistics of markables that are either reduced or ignored from the preprocessing of ARRAU^{RST} to convert it into a format consistent with the ICoref model used for the other datasets in this work.

Table 5.5 shows the number of training examples that were considered for each dataset. Datasets are shuffled once initially, so larger training sets are always a superset of a smaller one.

5.2.3.4 Source models

Since decoding in ICoref is greedy, it has three *learned* components: an encoder, a mention scorer, and a mention linker. This can be split into experiments where only the encoder is initialized and where the full model is initialized.

Dataset	# Training examples
OntoNotes^{zh}	[0, 10, 25, 50, 100, 250, 500, 1810]
OntoNotes^{ar}	[0, 10, 20, 40, 80, 160, 359]
PreCo	[5, 10, 25, 50, 100, 250, 500]
LitBank	[5,10, 20, 40, 80]
QBCoref	[5, 15, 30, 60, 120, 240]
ARRAU^{RST}	[10, 20, 40, 80, 160, 335]
SARA	[10, 20, 40, 80, 138*]
SemEval^{ca}	[10, 25, 50, 100, 250, 829]
SemEval^{es}	[10, 25, 50, 100, 250, 875]
SemEval^{it}	[10, 20, 40, 80]
SemEval^{nl}	[10, 20, 40, 80, 145]

Table 5.5: Training set sizes considered for each dataset. *For SARA, the entire fold is used, which contains 138 documents on average.

Pretrained encoders

For these models, only the encoder is initialized with a pretrained one and the rest of the model is randomly initialized. Joshi et al. (2020) trained the SpanBERT encoder on a collection of English data with a span boundary objective aimed at improving *span* representations. In addition, they finetune SpanBERT by training a coreference resolution system on OntoNotes (Joshi et al., 2019), which they release separately. This finetuned encoder is referred to as SpanBERT-On. Conneau et al. (2020) trained XLM-R, a cross-lingual encoder, on webcrawled text in 100 languages. As demonstrated in Section 5.1.2, it is effective at cross-lingual transfer, including coreference linking. In general, the "large" sizes of each model are used, except for one experiment with the "base" size of SpanBERT-on.

Trained models

Alternatively, the full model can be initialized with prior checkpoints. TRANSFER (ON) is a model downloaded directly from Xia et al. (2020a). I also train models on PreCo with SpanBERT-large (TRANSFER (PC)) and on OntoNotes^{en} with XLM-R (TRANSFER (EN)).[10] A variant of each model is also trained with gold mention boundaries, which skips the mention scorer.

5.2.4 Experiments and Results

For a single source model and target dataset, I train several models using a different number of input training examples described in Table 5.5. Coreference is evaluated with the average F_1 between MUC, B^3 and $CEAF_{\phi_4}$.[11]

Training Details

The hyperparameters are mostly the same as those of Xia et al. (2020a): $k = 0.4$ to select the top $0.4n$ spans; learning rates of 2e-4 for training the non-encoder parameters (with Adam); 1e-5 for the encoder (with AdamW); gradient clipping of 10; training up to 100 epochs with patience of 10 for early stopping based on dev F1. For all models, the full encoder is fine-tuned (except for Section 5.2.4.4). The max span width is 10 for SARA, 15 for PreCo and ARRAU^{RST}, 20 for LitBank and QBCoref, and 30 for all

[10]Cross-lingual models are trained separately because XLM-R and SpanBERT use different tokenizations.

[11]We score exact match for SARA (following prior work).

other datasets. These choices are made based on prior work or the statistics of the training set.

Each model was trained on a single 24GB Nvidia Quadro RTX 6000s for between 20 minutes to 16 hours, depending on the number of training examples. Due to the cost of training over 500 models, each model was trained only once. The English models use 373M parameters, of which 334M is the SpanBERT-large encoder. The multilingual models use 599M parameters, of which 560M is XLM-R large.

5.2.4.1 How effective is continued training for domain adaptation?

Continued training

Figure 5.3 shows that it is always beneficial to perform continued training on a source model, even if there is a large amount of target data. However, intuitively the differences are most pronounced in low-resource settings (with 10 fully-annotated documents) where it is still possible to adapt a strong model to perform non-randomly. These conclusions for coreference are similar to those drawn by Gururangan et al. (2020) on the effectiveness of domain- and task- pretraining of encoders for language classification tasks. These findings also support the intuition used by Urbizu et al. (2020), who choose PreCo as a pretraining corpus for ARRAU.

Continued training (and finetuning) is a core component of most NLP models,

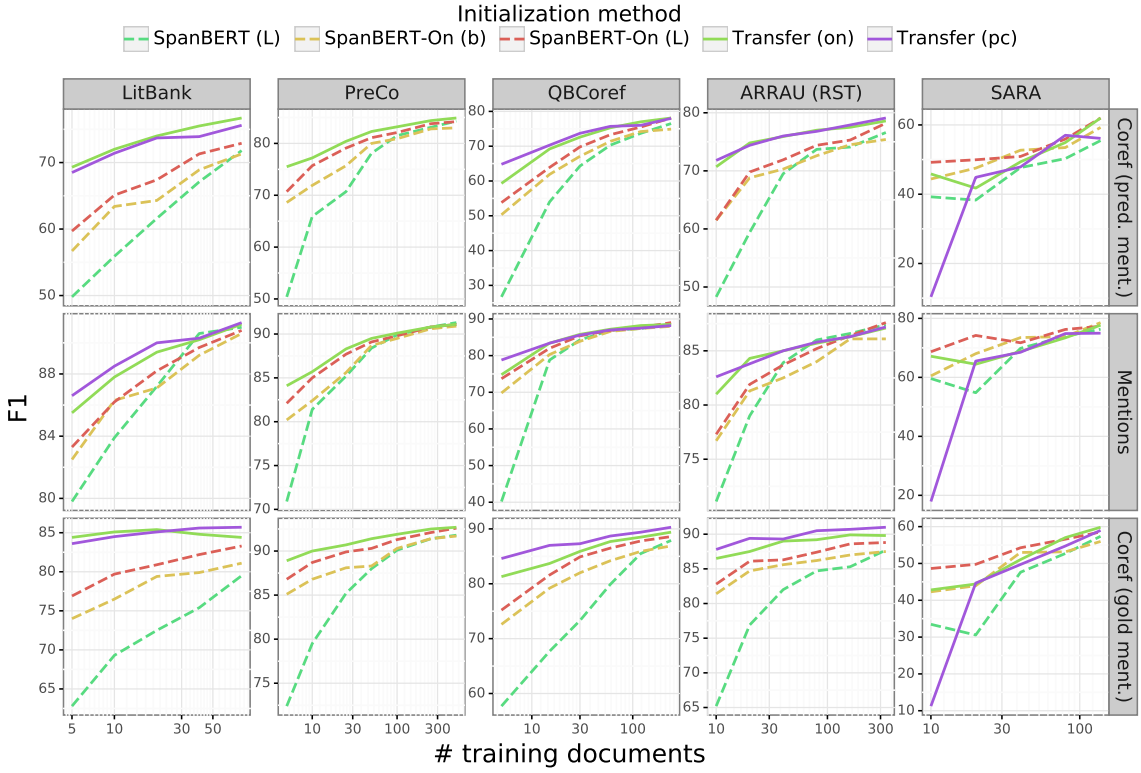


Figure 5.3: Each subplot shows the test performance for each model and (English) dataset when trained with a different number of documents. The first and second rows are coreference and mention boundary F_1 in the end-to-end setting, while the third row is the coreference F_1 with gold mentions. SpanBERT is a pretrained encoder, while the SpanBERT-On encoders are further finetuned on OntoNotes by Joshi et al. (2020), with **b**ase and **L**arge designating its size. Unlike these (dashed lines) models for which we initialize the encoder, the Transfer models (solid lines) use continued training and initialize the full model with one that has already been trained on a source dataset, either OntoNotes (on) or PreCo (pc).

as text embeddings are typically derived from large pretrained models. Joshi et al. (2018) find that model adaptation with contextualized word embeddings only requires a small set of partial annotations in the new domain for syntactic parsing. Meanwhile, Brown et al. (2020b) and Schick and Schütze (2021) find that pretrained language models can effectively learn a broad suite of sentence-level understanding, translation,

and question-answering tasks with just a few examples. This study corroborates their findings for a document-level information extraction task, since these models, based on strong pretrained encoders, perform well with just 5 or 10 training documents.

OntoNotes vs. PreCo

This study also finds that OntoNotes (Transfer (on)), despite being the benchmark dataset, is on par (or worse) as a pretraining dataset compared to PreCo (Transfer (pc)). One possibility is that because PreCo annotates for singletons, it is closer to the target datasets that also annotate singletons. This is evident when we compare the mention detection accuracy of the two models in low-data settings (e.g. LitBank or QBCoref at 5 examples). In the next setting, all models are given gold mention boundaries in pretraining, continued training, and testing, which would effectively evaluate just the linker. However, in this setting, PreCo outperforms OntoNotes even more on QBCoref, LitBank, as well as ARRAU^{RST}. This suggests PreCo as a preferred pretraining dataset over OntoNotes when there are few annotated documents.

Model size and pretraining

The publicly available models use the "base" and "large" encoders. While there are even larger encoders, coreference models using them are rare. For future model development, one may decide between using a publicly available small model and

retraining a large one from scratch. To simulate this, I compare a small encoder finetuned on OntoNotes, SpanBERT-On (b), with SpanBERT (L), which has not been trained on the task. This is also a realistic setting if there are hardware or compute limitations.

In all datasets, there is benefit to having some pretraining. When there is not much training data, the smaller (finetuned) encoder outperforms the larger encoder without finetuning. However, with enough data, the large model appears to surpass the smaller model. Nonetheless, there exist scenarios where continued training of a smaller model is desirable.

New benchmarks

Table 5.6 shows the test scores of our best model compared to prior work. For PreCo, we directly evaluate on the fully-trained model without continued training, as the full dataset is sufficiently large. Since some of these datasets are understudied, these are primarily intended as stronger baselines for future work.[12] The purpose is to quantify the effectiveness of continued training and highlight PreCo as an alternative pretraining dataset. Note that this strong performance is achieved without hyperparameter tuning or incorporating any language or domain specific features.

For Table 5.6, the score of the best model between Transfer (on) and Transfer (pc) is reported, based on their dev scores on each dataset. These are listed below in

[12]Contemporaneous and subsequent work has since established even stronger baselines for several of datasets, e.g. LitBank (Thirukovalluru et al., 2021).

Dataset	Prior work	Previous Model	Previous Score	Our best	Our Model
PreCo	Wu and Gardner (2021)	SpanBERT + C2F	85.0	**88.0**	PC
LitBank	Thirukovalluru et al. (2021)	SpanBERT + C2F	**78.4**	76.7	ON
QBCoref	Guha et al. (2015)	Berkeley	< 35	**78.1**	ON
ARRAU^{RST}	Yu et al. (2020b)	BERT + cluster ranking	77.9	**79.1***	PC
SARA	Holzenberger and Van Durme (2021)	string match baselines	55.1	**72.9**	ON
OntoNotes^{zh}	Chen and Ng (2012)	Multi-pass sieve	62.2	**69.0**	EN
OntoNotes^{ar}	Aloraini et al. (2020)	AraBERT + C2F	**63.9**	58.5	EN
SemEval^{ca}	Attardi et al. (2010)	feature-based + MaxEnt	48.2	**51.0**	EN
SemEval^{es}	Attardi et al. (2010)	feature-based + MaxEnt	49.0	**51.3**	EN
SemEval^{it}	Kobdani and Schütze (2010)	feature-based + decision tree	**60.8**	36.7	EN
SemEval^{nl}	Kobdani and Schütze (2010)	feature-based + decision tree	19.1	**55.4**	EN

Table 5.6: Test F_1 on all datasets and the previous state-of-the-art on each dataset, to the best of our knowledge. Again, the goal is to benchmark the general method of continued training described in this study, which will not necessarily outperform models that incorporate domain or language specific knowledge. The best TRANSFER model is determined by the dev set. *ARRAU^{RST} is not directly comparable to prior work as it is test on a slightly differently-preprocessed subset. Multi-pass sieve (Raghunathan et al., 2010), Berkeley (Durrett and Klein, 2013), and C2F (Lee et al., 2018) refer to widely-used coreference resolution models.

Table 5.7.

Dataset	ON	PC	EN
PreCo	82.4	**85.2**	-
LitBank	**77.3**	76.3	-
QBCoref	**79.1**	78.7	-
ARRAU^{RST}	77.7	**79.3**	-
SARA	**77.7**	75.4	-
OntoNotes^{zh}	-	-	**69.0**
OntoNotes^{ar}	-	-	**62.3**
SemEval^{ca}	-	-	**51.4**
SemEval^{es}	-	-	**52.1**
SemEval^{it}	-	-	**36.1**
SemEval^{nl}	-	-	**48.3**

Table 5.7: Dev. F1 scores on each of the models and datasets presented in Table 5.6. For the English dataset, the test score of the model with the best performing score is reported in Table 5.6.

Cross-lingual transfer

Figure 5.4 shows the results for multilingual coreference resolution. The gap in performance at low-data conditions (and the high initial starting point) shows that transfer via continued training is also effective cross-lingually in the end-to-end document-level task. These results corroborate prior work (Conneau et al., 2020) by providing more evidence for XLM-R's cross-lingual transfer ability, in this case on the full end-to-end task. Given these results, *joint* multilingual pretraining followed by continued training might be an even more effective recipe for creating the best models for each language. This is out of scope for this study, which is focused on transfer from single datasets.

5.2.4.2 How to allocate annotated documents?

In Figure 5.3, the experiments for each dataset used the same dev set for model selection to improve comparability. At the same time, I observe that adding even a few more training examples can lead to improved performance. For some datasets, like PreCo, the size of the dev set used for model selection in our experiments greatly outnumbers the number of training documents. Next, I explore allocating fewer documents for model selection.

20 models for PreCo are trained with a different number of examples using SpanBERT-On (L) and Transfer (On). Each model is trained for 60 epochs and makes predictions on all 500 dev examples. Next, for each dev set size, a subset of

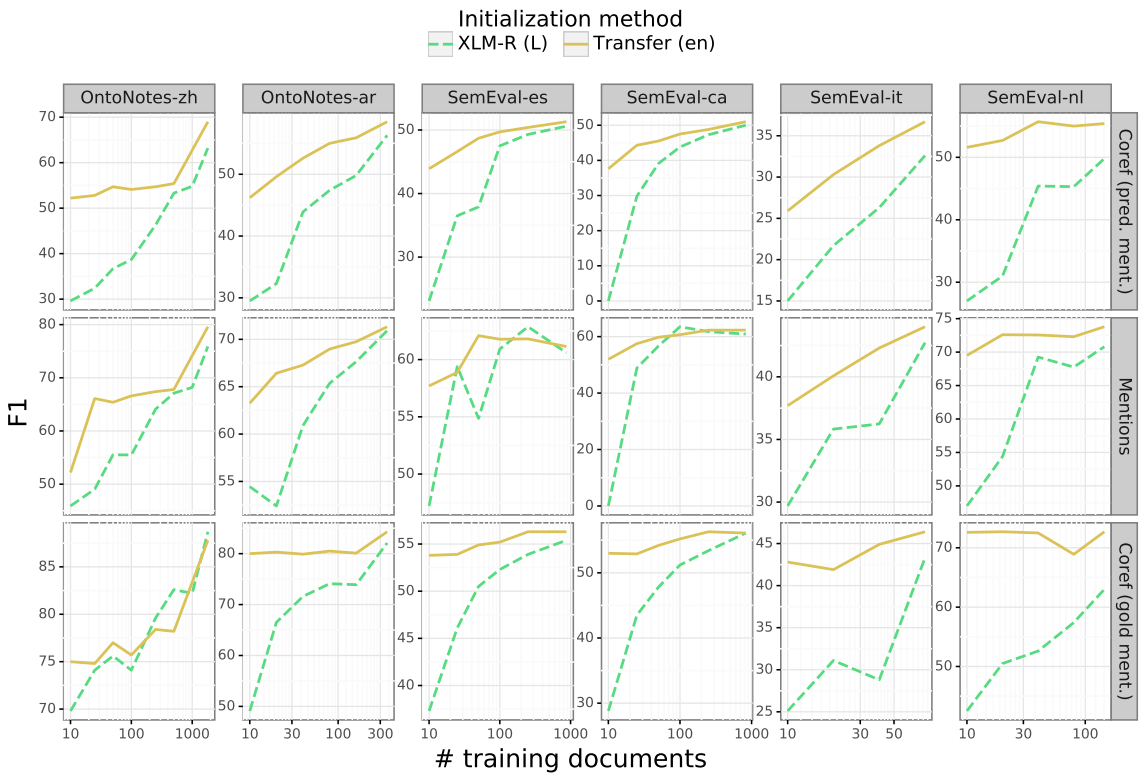


Figure 5.4: Like Figure 5.3, this plot demonstates the effectiveness of continued training across different *languages*. XLM-R uses a pretrained encoder (dashed line), while TRANSFER (EN) is first trained on OntoNotes^{en} (solid line).

the full predictions is sampled and used to determine, post-hoc, the checkpoint at which the model would have stopped had that sampled subset instead been the dev set. This is performed 20 times, yielding 20 such subsets, which is used to compute the *expected* scores and standard deviation for each model, along with how frequently the subset agreed with the full dev set.

Figure 5.5 summarizes the results, showing remarkable stability in expectation even with tiny dev sets, often less than a couple points behind using the full dev set. Given a fixed budget of documents or annotations, these results suggest that it is

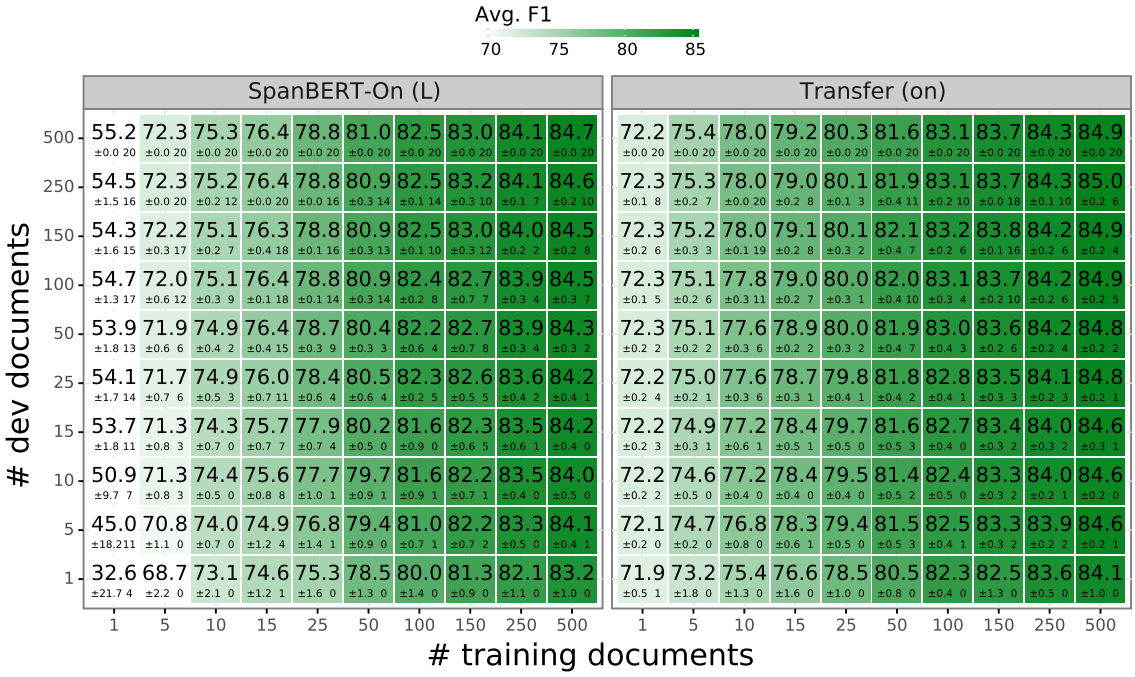


Figure 5.5: The expected test F_1 (and standard deviation) on the PreCo dataset for a given number of training documents and 20 sampled subsets of dev documents for two models described in Section 5.2.3.4. The number of runs matching the best full dev checkpoint is in the lower-right. We find that the dev set size has relatively little impact.

beneficial to allocate as many documents as possible towards training, leaving behind a small set for model selection.

5.2.4.3 How much do the source models forget?

To measure the degree of catastrophic forgetting (McCloskey and Cohen, 1989), the source datasets of each TRANSFER model is revisited and its performance is measured in the presence of more training data.[13] In Figure 5.6, it is evident that on some datasets, the performance drop is especially pronounced after training on just

[13]For datasets with k-folds, the mean across folds is reported.

10 examples in the target dataset.

I hypothesize that this is due to easy-to-learn changes between the annotation guidelines that are incompatible between the two datasets, like the annotation of certain entity types. Two pairs, (OntoNotes^{en}→OntoNotes^{zh}) and (PreCo→ARRAU^{RST}) are less affected by continued training. For OntoNotes, the same guidelines are used for all languages. Meanwhile, PreCo and ARRAU^{RST} are more similar in annotation guidelines than any other pair since they both include singletons. On the other hand, (OntoNotes→ARRAU^{RST}) shows a substantial drop in performance despite the two datasets containing overlapping documents.

In the cross-lingual setting, the drops are smaller than across English datasets. This could be due to several factors. The XLM-R encoder is already trained multilingually and has strong crosslingual performance (Conneau et al., 2020), while English encoders are not well-suited for all domains, like law (Chalkidis et al., 2020). The crosslingual datasets in this study (OntoNotes and SemEval) are primarily in the same domain (newswire) and share similar annotation guidelines. And, in some cases where the trend looks flatter (SemEval^{it}, Semeval^{nl}, and even SARA), the training dataset is also smaller.

5.2.4.4 Which encoder layers are important?

Training the entire encoder is an expensive cost of continued training, both in terms of training time and in the number of *new* parameters introduced by a new

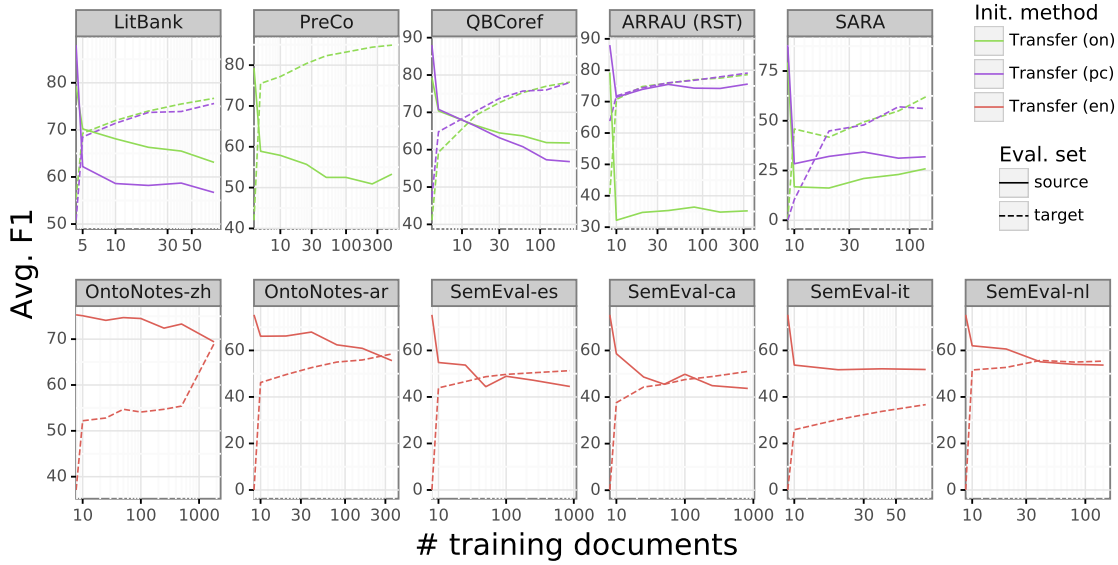


Figure 5.6: Average F_1 of the models on both the target and the original datasets as different number of (target) training examples are used in continued training. The dashed lines are the scores on the target dataset (mirroring Figure 5.3) while the solid lines show performance on the original dataset.

target dataset. I experiment with freezing some parameters of the encoder and training the top-k layers, along with the rest of the model, for each of the "large" encoders. This is motivated by Section 5.1.2, prior work which uses just the top four layers (Aloraini et al., 2020), and by findings from encoder probing that higher layers are more salient for coreference (Tenney et al., 2019a). I explore this question on three datasets (LitBank, QBCoref, OntoNotes^{zh}).

Figure 5.7 shows that there are gains to training some layers, but it is not always necessary to train the full model. In particular, for transferred models, unfreezing more layers of the encoder could even lead to worse performance. On the other hand, untrained models generally benefit from training more of the encoder. These trends are

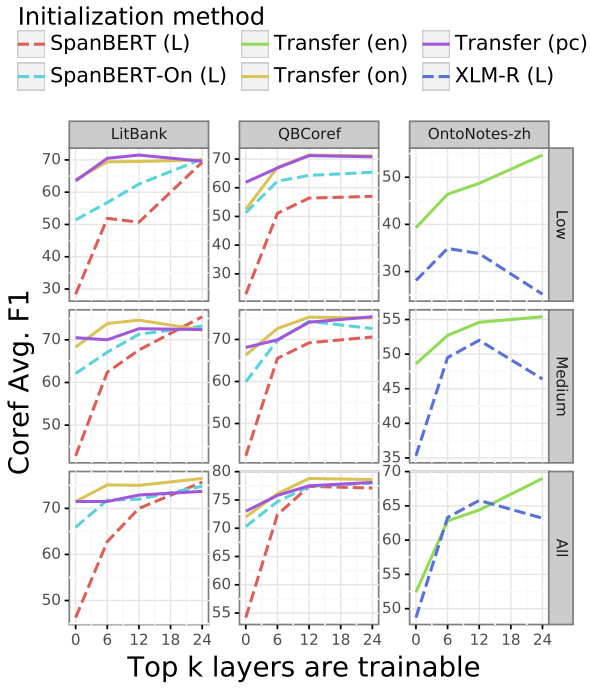


Figure 5.7: Average F_1 across different models and number of trainable layers, varying between 0, 6, 12 or 24 layers. *Low* vs. *Medium* vs. *All* describes the number of documents used for the first fold of LitBank (10, 40, 80 documents), QBCoref (15, 60, 240 documents), and OntoNotes^{zh} (50, 500, 1810 documents). The initialization methods follow those used throughout the paper.

observed in both datasets and data quantities. This is also observed for OntoNotes^{zh} and in medium data conditions.

5.2.5 Conclusion and Limitations

Section 5.2 comprehensively examines the transferability of neural coreference resolution models. I explore several model initialization methods across a wide set of domains and languages, and with a different number of training examples, to demonstrate the universal effectiveness of continued training. Additionally, this

method results in improved performance over prior work on many of these datasets. Furthermore, I find that PreCo can be effectively used for pretraining, suggesting a viable alternative to OntoNotes.

Upon further analysis, I find that: given a fixed number of annotated documents, few need to be allocated for model selection; continued training also suffers from catastrophic forgetting; and continued training is effective with partially frozen encoders. This study and its set of benchmarks serve as a reference for future work in coreference resolution model adaptation, especially for scenarios where annotation can be expensive or data may be scarce.

Chapter 6

Reducing model size

This chapter first describes a brief overview of neural model compression and pruning methods. While it is not aimed to be a comprehensive survey, it provides additional context for my subsequent contributions. Section 6.2 subsequently describes a general study and method for pruning *multitask* models. The goal is two-fold. First, I want to create a single model that can perform well across multiple tasks (akin to the multilingual attempt in Section 5.1.2 or multi-dataset attempt in Toshniwal et al. (2021)). However, I also wish to address the relatively large size of the encoder (relative to the parameter count for mention detection and linking). I explore a method for simultaneously training a multi-task model while compressing. In Section 6.3, I apply this method to ICoref as a proof of concept, showcasing one possible method for reducing the size of multi-dataset coreference resolution models.

6.1 A brief overview of model compression

The goal of model compression is to start with a large model and produce a smaller model without substantially affecting accuracy, typically with the goal of improving inference speed or (on-disk) model size. There are a few broad classes of methods for doing so: knowledge distillation, model pruning, factorization techniques, and quantization. This is not an exhaustive set; more in-depth discussion can be found in several comprehensive surveys on model compression (Cheng et al., 2017; Deng et al.,

2020; Xu and McAuley, 2022).

In knowledge distillation (Bucila et al., 2006; Hinton et al., 2015), a larger model (or ensemble) is used to "teach" a smaller model. Given a large model M_{large} and labeled data (X, Y), a smaller model M_{small} can use both the label y and the soft labels (e.g. logits scores or intermediate attention weights) $M(x)$ as part of its training signal. This general purpose machine learning technique has been applied to a variety of models and architectures in NLP (Kim and Rush, 2016; Sanh et al., 2020a).

Model pruning refers to removing unimportant weights in a model by setting them to zero (LeCun et al., 1990). There are several methods for determining the importance of the weights: directly using the magnitude of the weights (Han et al., 2015; Frankle and Carbin, 2019), including a regularization loss on individual or groups of weights (Murray and Chiang, 2015; Alvarez and Salzmann, 2016), or learning importance scoers (Sanh et al., 2020b). Generally, after pruning, the model needs to be trained further to recover performance. If the pruning is performed in a *structured* manner by dropping out entire attention heads, layers, or columns (Michel et al., 2019; Fan et al., 2019; Murray et al., 2019), then this would also result in a substantially smaller model in practice. On the other hand, pruning can also be *unstructured*; in this case, the model might not shrink in size as neither the dimensions of the matrices nor the computation graph shrink in size (See et al., 2016). Thus, this would only lead to speedups in practice with fast implementations of sparse matrix operations.

Another approach includes matrix or tensor factorization, which aims to

reparameterize matrices of size with low-rank approximations by using singular value decomposition (Sainath et al., 2013; Haeffele et al., 2014). This results in overhead at inference, but can substantially reduces the number of parameters of the model if the rank is sufficiently reduced.[1] This has been used for several neural architectures, like Transformers (Ben Noach and Goldberg, 2020), LSTMs (Grachev et al., 2017) and CNNs (Idelbayev and Carreira-Perpinan, 2020).

Finally, quantization is a different approach which still keeps the number of parameters fixed but reduces the number of bits for each `float` within the weights (for example, switching from 32-bit floating point to 8-bit integers) (Micikevicius et al., 2018; Jacob et al., 2018). This shrinks the effective size of the model in terms of number of bytes. Some of the main challenges are correctly computing gradients during training and ensuring speedups on real hardware. Subsequent work has applied quantization to models in NLP (e.g. Shen et al. (2020) or Bai et al. (2021)) and is often used in conjunction with other methods.

In this chapter, I take a deeper dive into model pruning as it best supports the paradigm of continued training established in Section 5.2. Specifically, I build on a pre-existing pruning method (Sanh et al., 2020b) that is geared towards fine-tuning (encoders) on a new task. However, all of the methods described are complementary, and are often combined when optimizing for the smallest and fastest model (e.g. (Kim and Hassan, 2020)).

[1] A more detailed explanation is in Section 6.2.2.1.

6.2 Multitask Model Pruning

Note

This was presented as **Pruning Pretrained Encoders with a Multitask Objective** at the NeurIPS 2021 Workshop on Efficient Natural Language and Speech Processing (ENLSP).[2]

Abstract

The sizes of pretrained language models make them challenging and expensive to use when there are multiple desired downstream tasks. This work adopts recent strategies for model pruning during finetuning to explore the question of whether it is possible to prune a single encoder so that it can be used for multiple tasks. I allocate a fixed parameter budget and compare pruning a single model with a multitask objective against the best ensemble of single-task models. Under two pruning strategies (element-wise and rank pruning), the approach with the multitask objective outperforms training models separately when averaged across all tasks, and it is competitive on each individual one. Additional analysis finds that using a multitask objective during pruning can also be an effective method for reducing model sizes for low-resource tasks.

[2]The work was done in 2021 with Richard Shin.

6.2.1 Introduction and Background

In the last few years, NLP models have relied on pretrained text encoders like BERT (Devlin et al., 2019), which perform well when finetuned across many downstream NLP tasks. At the same time, these models are often overparameterized for the downstream task, leading to a surge of interest in reducing encoder size while retaining most of its performance on downstream tasks (Sun et al., 2020b; Sanh et al., 2020a).

Meanwhile, there has been interest in adapting a *single* model to *multiple* downstream tasks through the use of a small number of additional, task-specific parameters (Houlsby et al., 2019; Shin et al., 2020; Hu et al., 2021). These techniques are useful for efficiently sharing large base models during training by freezing the underlying encoder and finetuning parameters dependent on the target task. Switching between tasks is also cheap: only a small component is changed.[3] These methods for extending models of size N with only ϵ parameters per task can perform well on t tasks at the cost of $N + t\epsilon$ instead of tN parameters, and architectural innovations have led to smaller ϵ (Karimi Mahabadi et al., 2021a; Karimi Mahabadi et al., 2021b). In practice, this allows for deployment of models for multiple tasks at the cost of a single model in terms of memory or disk space.

This section describes the possibility of further reducing the number of parameters used by these multitask models to be substantially smaller than N. Specifically, this is done by pruning *multitask* models using a multitask training objective. To get *all*

[3]These can be viewed as a more expensive version of the genre embedding features in OntoNotes coreference resolution models.

of the benefits described above, the goal is a substantially pruned model that also performs well on multiple tasks. In addition, I ask whether performance on individual task can be improved by leveraging data from the other tasks, which is a strategy employed by general-purpose language modeling (Liu et al., 2019; Aghajanyan et al., 2021).

This is an empirical study specifically for pruning in the multitask scenario, which contributes:

- An extension of both structured and unstructured pruning methods to the multitask setting.
- Under both methods, findings that suggest a multitask model consistently outperforms a combination of single-task models for a given, fixed budget.
- A multitask objective does not necessarily lead to a loss in performance on any individual task. In some cases, it enables improved performance for tasks with smaller dataset sizes.

6.2.2 Approach

In model pruning, elements of a weight matrix are *pruned* or set to zero. This is usually determined using a heuristic such as a fixed magnitude threshold or relative top-k cutoff which is gradually increased during training on the task dataset. The

objective is to create a model with high *sparsity*,[4] which measures the fraction of removed (or zeroed out) weights.

6.2.2.1 Pruning Methods

I explore 8 settings (Table 6.1) for model pruning: what gets pruned (element-wise, unstructured pruning vs. rank, structured pruning), varying pruning method (magnitude vs. movement), and uniformity of pruning (global vs. local).

Variable	Choices	How it changes pruning objective
Structure	Element-wise	Prune parameters independently from each other
	Rank	SVD then prune diagonal entries (rank)
Selection	Magnitude	Remove model weights with smallest magnitudes
	Movement	Learned importance scores per weight
Scope	Global	top-k computed across entire model
	Local	top-k computed per weight matrix

Table 6.1: The main variable in our pruning comparison experiments is structure, although we also explore two other variables. There are 3 variables with 2 choices each, leading to $2^3 = 8$ combinations for model settings.

Unstructured vs. structured pruning

In prior work for model pruning, individual elements of a weight (matrix) are typically pruned separately, or *element-wise*, in an unstructured manner, which can lead to inconvenient sparsity patterns for a single weight matrix. This pattern means that it can be difficult to realize real-world gains in efficiency on typical hardware, particularly on GPUs (Gale et al., 2020). As such, other work considers structured

[4]Equivalently, low density.

pruning of entire rows or columns of the matrices, which makes it much easier to realize efficiency gains (Fan et al., 2021; Lagunas et al., 2021).

Both the theoretical setting and the real-world setting are important towards achieving efficiency in practice. Thus, an alternative structured pruning approach is explored: *rank pruning* (Yang et al., 2020).

Starting with a weight $\mathbf{W} \in \mathbb{R}^{m \times n}$, SVD lets us approximate $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$ where $\mathbf{U} \in \mathbb{R}^{m \times k}$, $\mathbf{\Sigma} \in \mathbb{R}^{k \times k}$ is diagonal, and $\mathbf{V} \in \mathbb{R}^{k \times n}$. Initially, $k = \min(m, n)$. By pruning values from the diagonal of $\mathbf{\Sigma}$, we reduce its rank from k to k'. Consequently, entire columns in $\mathbf{U}$ and rows in $\mathbf{V}$ can be pruned, resulting in three reduced matrices: $\mathbf{U}'$, $\mathbf{\Sigma}'$, and $\mathbf{V}'$.

The resulting matrices can be stored as $\mathbf{U}'\mathbf{\Sigma}' \in \mathbb{R}^{m \times k'}$ and $\mathbf{V}' \in \mathbb{R}^{k' \times n}$, which uses $k'(m + n)$ parameters. If $k' < \frac{k}{2}$, this is smaller than the original $m \times n$ size of $\mathbf{W}$.[5]

Similarly, decomposing $\mathbf{W}$ and pruning in this way would lead to faster inference – for an input $x \in \mathbb{R}^{m \times l}$, $\mathbf{W}^\top x$ costs lmn scalar multiplication operations while

[5] If $k' > \frac{k}{2}$, we can recover the unfactorized form, $\mathbf{W}' = \mathbf{U}'\mathbf{\Sigma}'\mathbf{V}'$. Thus, rank pruning would never use more parameters or computation than element-wise pruning. For a single weight matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$, the original weight matrix has size mn.

After performing SVD, $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$. After pruning to a final rank k', the reduced $\mathbf{U}'$ has $k'm$ elements and $\mathbf{V}'$ has $k'n$ elements, and so storing the decomposed matrix costs $k'm + k'n$.

When $k' < \frac{\min(m,n)}{2}$, then

$$k'm + k'n < \frac{\min(m, n)}{2}(m + n) \tag{6.1}$$

$$\leq \frac{\min(m, n)}{2} \cdot 2 \cdot \max(m, n) \tag{6.2}$$

$$= \min(m, n) \max(m, n) = mn. \tag{6.3}$$

Note that this bound is not tight, and when $m \neq n$, k' can be higher. This occasionally turns out to be the case in the feedforward layers of BERT. Nonetheless, this bound is sufficient for all m, n and is the one used here.

$((\mathbf{U}'\mathbf{\Sigma}')\mathbf{V}')^\top x$ only needs $lk'(m+n)$.

The elements in $\mathbf{\Sigma}$ are initially the singular values of $\mathbf{W}$, and so their magnitudes are interpretable as the importance of a particular dimension. Yang et al. (2020) prune directly on the magnitudes and use orthogonal regularization to constrain $\mathbf{\Sigma}$ to be close to the singular values. In contrast, SVD is only used as initialization without additional regularization. Thus, interpreting the weights as singular values may not be valid after finetuning, and so there may be better selection heuristics than magnitude pruning.

Magnitude vs. movement pruning

Two pruning selection heuristics are considered. *Magnitude pruning* is a well-studied method which incrementally removes (sets to zero) the smallest-magnitude weights during training, until only the top-k largest weights remain at completion (LeCun et al., 1990; Han et al., 2015; Zhu and Gupta, 2018). In this section, k refers to a *fraction* of the weights, and so it directly controls the size of the pruned model.

Movement pruning (Sanh et al., 2020b) is a first-order method which prunes by using learned importance scores that correspond to the cumulative change during finetuning of each weight matrix element, effectively retaining weights moving away from 0. Formally, for a weight $\mathbf{W}$, this method associates with it a set of scores $\mathbf{S}$. In the forward pass, a binary mask $\mathbf{M} = \text{Top}_k(\mathbf{S})$ is applied to the weight, and so for an input $\mathbf{x}$, $(\mathbf{W} \odot \mathbf{M})\mathbf{x}$ is used instead, where $\odot$ is the element-wise product. In

the backward pass, all weights are updated as Top_k is ignored and its gradient is approximated using a straight-through estimator (Bengio et al., 2013). Sanh et al. (2020b) showed this to be effective for finetuning a pretrained model using a new objective.

There are extensions to both methods which use an minimum threshold rather than top-k; however, controlling the sparsity would then requires a threshold sweep per task to discover the best threshold, which is expensive especially in the multitask setting. For simplicity, this study directly controls k.

Global vs. local pruning

Finally, pruning can be either *local* (each weight matrix is pruned to the same sparsity) or *global* (only the entire model needs to hit a target sparsity). Pruning globally allows for more aggressive pruning of less useful model components or layers, assuming weights are globally comparable.

6.2.2.2 Multitask extension

Extending these methods to the multitask setting is straightforward. Separate, unpruned classification heads for each task can be learned while each task model shares a common set of pruned encoder weights (and learned importance scores, in the case of movement pruning) across all tasks.[6] In multitask pruning, each task is

[6]One could also selectively prune weights (and learn importance scores) based on the task (Liang et al., 2021). However, I found that doing so results in about the same or worse task performance while incurring a higher parameter cost.

sampled uniformly at random and optimized for that task's objective.

6.2.2.3 Model and Datasets

Like prior work (Jiao et al., 2020; Sun et al., 2020b; Sanh et al., 2020b), I prune the BERT-base model (Devlin et al., 2019). The reported pruning fraction is relative only to the 12-layer Transformer. The embedding and output classification layers are considered a fixed cost. This is also the treatment in Sanh et al. (2020b). The single-task baselines use their best hyperparameters for all main experiments.

Specifically, for the single-task baseline models, the best hyperparameters from Sanh et al. (2020b) are used. For rank pruning the learning rate of $\mathbf{\Sigma}$ is increased to 5×10^{-3} after a small search. The multi-task models are trained for 8 epochs with 2 initial and 2 final warm-up epochs. This is close the average of the number of epochs used for the three separate tasks.

Like Sanh et al. (2020b), English language understanding datasets act as the primary benchmark: MNLI (Williams et al., 2018), SQuADv1.1 (Rajpurkar et al., 2016), and QQP (Iyer et al., 2017). Since the goal in this section is to evaluate the compressibility of a multitask model, there are no claims regarding specific *tasks*.

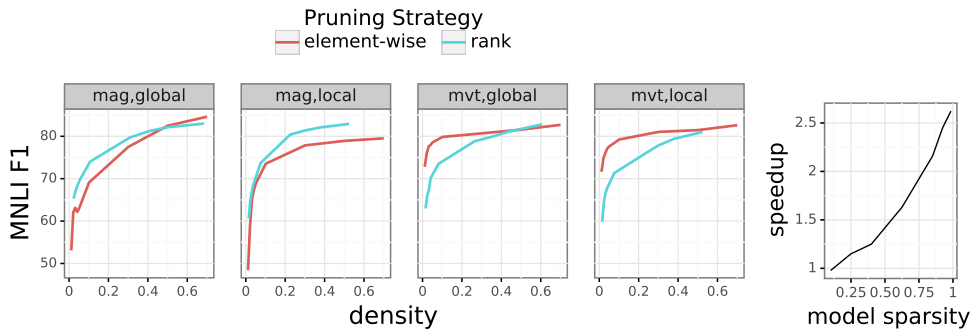


Figure 6.1: Left: The performance on MNLI (dev) across [magnitude, movement] and [global, local] pruning strategies. Within each plot, we show the performance of pruned element-wise and rank-pruned models. Right: Comparison of the runtime of a model using rank pruning relative to an entirely dense model, showing that the structured sparsity ensured by rank pruning can lead to practical benefits. *Density* is 1 - sparsity.

6.2.3 Experiments and Results

6.2.3.1 Comparison of pruning methods

First, I establish the relative task performance difference between element-wise and rank pruning and confirm that rank pruning (a structured approach) outperforms element-wise pruning (an unstructured approach) in practice. BERT-base is pruned to various target sparsities while finetuning on MNLI for each of the parameter combinations described in Section 6.2.2.1. This is done in the single-task setting to compare the possible benefits that can be derived from rank pruning and to find the best configuration for both methods.

Task performance

The results in element-wise pruning once again demonstrate the superiority of global movement pruning (Sanh et al., 2020b). In contrast, for rank pruning, local magnitude pruning is generally the best strategy. The poor performance of global pruning can be explained by the fact that it is possible to prune some parameters marginally (i.e. not beyond the $\frac{k}{2}$ threshold needed to see improvements) and so some of the budget allocated towards pruning is unused. Furthermore, the magnitudes of the singular values of the feedforward and self-attention layers are not necessarily comparable, as the attention layers were found to be pruned more aggressively.

Comparing between the best settings for unstructured (element-wise) and structured (rank) pruning, it appears that unstructured pruning retains performance better when the parameter count is low.

Runtime comparison

In Figure 6.1, I make a simplifying assumption that unstructured pruning does not shrink the size of any individual matrix, and so the runtime at any sparsity would be equal to that of an unpruned model. Under that assumption, rank pruning offers substantial speedups and is almost immediately (at around 10% sparsity) faster than the dense model.

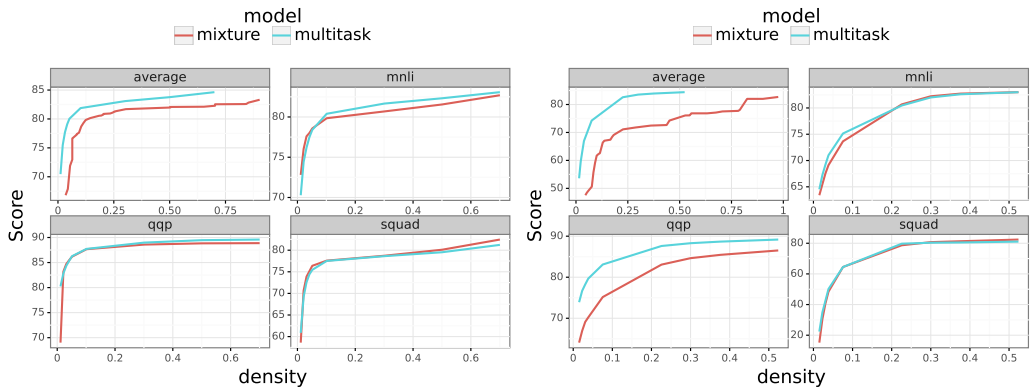
However, sparsifying a matrix can lead to specialized hardware and algorithmic optimizations as demonstrated by sparse multiplication libraries (Gale et al., 2020).

Lagunas et al. (2021) optimize element-wise unstructured pruning in a simple manner by removing entirely pruned rows, columns or attention heads. They show that even at high sparsities (more than 90%), this strategy achieves at most around a 1.5× speedup. Meanwhile, rank pruning achieves a comparable speedup at just 50% and almost a 2.5× speedup at 90%. While rank pruning has lower F1 under a fixed parameter budget, it is a competitive option given a fixed *latency* budget.

Global rank pruning

One limitation of global rank pruning is that the initialization is not normalized across weights. Specifically, with *magnitude* as the selection heuristic, attention weights are pruned more aggressively because most of their singular values are smaller than those of the feedforward parameters. This could result in important weights in the attention layers being pruned before less important ones in the feedforward layers, and explain why global rank pruning performs poorly at high sparsities (as shown in Figure 6.1a). The opposite behavior is observed for rank *movement* pruning, where pruning globally is slightly preferred over pruning locally. This suggests that first-order information might be more comparable globally.

Future work can investigate finding a balance between using the interpretability of the magnitudes of the singular values and first-order information from the finetuning process. Alternatively, one could explore a hybrid pruning strategy: with *magnitude* pruning, separate thresholds (or k) for the attention and the feedforward parameters.



(a) Unstructured (element-wise) pruning (b) Structured (rank) pruning

Figure 6.2: Given a fixed parameter budget (expressed as a fraction of a single model size), a single model pruned with a multitask objective (blue) is compared against the best combination of 3 individual task models for a given size. The red line ("mixture") is the Pareto frontier of these combinations.

6.2.4 Multitask pruning

With the above baseline established, the next experiments are in the multitask setting with MNLI, QQP, and SQuAD. The goal is to answer: given a fixed parameter budget for all tasks, is it better to train (and prune) three specialized models or one substantially larger multitask model for all three tasks?

Concretely, models are pruned to varying sparsities using four separate objectives: MNLI, QQP, SQuAD, and multitask. The best mixture of single-task models is compared to a single model pruned with a multitask objective. The best mixture is determined by the Pareto frontier of all possible collections of models. Each single-task model was pruned to [1, 2, 3, 4, 5, 10, 15, 30] percent and the Pareto frontier (i.e. the best mixture) was calculated based on dev. macro average F1 while single-task

Model	Rank	Prune %	MNLI 393K	SQuAD 87.6K	QQP 364K	QNLI 105K	SST-2 67K	MRPC 3.7K	STSB 7K	RTE 2.5K	CoLA 8.5K	Avg.
No pruning	–	100%	84.0	84.9	89.1	90.7	92.3	86.5	88.5	65.7	56.3	82.0
RP (3)	38	7.6%	73.8	64.3	83.1	-	-	-	-	-	-	-
RP (1)	38	7.6%	73.6	64.4	75.2	80.0	87.4	75.7	14.4	52.7	0	58.2
BNG	150	29.2%	-	-	-	-	91.3 ± 0.4	87.8 ± 0.6	-	-	38.7 ± 1.6	-
RP (9)	38	7.6%	72.1	54.2	81.3	84.7	87.6	83.6	86.3	67.1	23.6	71.2
RP (9)	76	15.0%	76.8	70.5	84.5	88.2	89.8	87.5	88.0	70.4	34.6	76.7

Table 6.2: Individual task performance (dev.) of a model pruned using rank pruning with the multitask objective. The size of the training set for each task is also listed. RP (9) is trained on all nine tasks, RP (3) is the three-task model from Section 6.2.4, and RP (1) represents 9 separate single-task baseline models. BNG (Ben Noach and Goldberg, 2020) is three separate single-task low-rank models tuned using knowledge distillation.

models were selected based on dev. F1. Figure 6.2 shows that the multitask model outperforms the mixture on a macro-averaged 3-task metric. In addition, it matches or exceeds the performance on individual tasks.

Multitask pruning outperforms the mixture with both element-wise pruning and rank pruning, suggesting that the ability to leverage a multitask objective during pruning may extend to other, novel methods for pruning.

6.2.5 Auxiliary multitask pruning objective

In Section 6.2.4, I observed that in some cases, a multitask pruning objective is helpful even when only one task (e.g. QQP) is of interest. This motivates a follow-up question: if we only care about a single task, should we still use a multitask objective?[7]

To test this hypothesis further, I expand the three tasks to 9 and prune a model

[7]This is related to similar themes from Section 5.1.2, where including additional non-Russian multilingual data can help Russian coreference performance.

with a 9-task objective, sampling each task uniformly at random. In addition to the three aforementioned tasks, this includes CoLA (Warstadt et al., 2019), SST-2 (Socher et al., 2013), MRPC (Dolan and Brockett, 2005), STSB (Cer et al., 2017), and QNLI (Rajpurkar et al., 2016), and RTE (Dagan et al., 2005).

Table 6.2 shows the performance on each of the tasks when pruning using this 9-task objective with local magnitude rank pruning. For comparison, a single-task model is also pruned with the same method. The multitask models (RP (9)) perform well on the smaller datasets of RTE, CoLA, and STSB, outperforming the single-task baseline (RP (1)) and they come close to prior models which are larger and also use a separate hyperparameter search for each task (Ben Noach and Goldberg, 2020). In contrast, this method in this setting required *no additional hyperparameter tuning* (beyond pruning heuristic decisions made in Section 6.2.3.1). These results collectively suggest that multitask-based pruning offers another way to effectively prune models for low-resourced tasks.

6.2.6 Discussion and Limitations

This section is limited to multitask compression, which aims to perform well on multiple tasks with a given parameter budget smaller than the size of single-task model. Under both unstructured (element-wise) pruning and structured (rank) pruning strategies, pruning with a multitask objective outperforms a combination of multiple models pruned separately on individual task objectives. Additionally, pruning with

a multitask objective (without additional hyperparameter tuning) can help further prune the model for tasks with smaller datasets. These lessons learned from this restricted setting can guide future work in compressing multitask models.

Specifically, it suggests a method for pruning coreference resolution models. Throughout the thesis, I demonstrate that models struggle to generalize across datasets and domains without additional in-domain data. We can view each dataset, domain, and language as a "task" under the framework of this multitask pruning method. Under this framing, a (small) multi-dataset coreference resolution model can be created and evaluated. This is explored next.

6.3 Reducing model size for multiple coreference datasets

This section applies the techniques from the previous section to coreference resolution models. I demonstrate a proof of concept approach for reducing model size and speculate on additional directions to reduce model size. Concretely, this experiment simultaneously contributes two rarely-studied areas of coreference resolution: a multi-dataset model with parameter sharing at the encoder-level and an attempt to reduce the size of these coreference resolution models.

6.3.1 Background

One model, many datasets

There is substantial prior work in creating models aimed at multiple target datasets for coreference resolution. Section 5.1.2 aggregates datasets from several languages to train a single multilingual model. In addition, Toshniwal et al. (2021) create a single model aimed at multiple English datasets from different domains. OntoNotes 5.0 itself is a collection of multigenre data (Weischedel et al., 2013), and modeling for this dataset typically use the document genre as an input to the model (e.g. Clark and Manning (2016b)). We could reinterpret this learned genre vector as "task-specific" parameters, like adapters (Houlsby et al., 2019), tuned prefix embeddings (Li and Liang, 2021), or even tuned bias terms (Ben Zaken et al., 2022).

From the perspective of encoder pretraining, some models use disparate tasks as a multitask training objectives and (Liu et al., 2019; Sun et al., 2020a). These model architectures share the the transformer encoder parameters while they have separate classifiers for each task or type of task. Going further, some models targeting multiple data distributions only share lower layers of encoders while the higher-up layer are specialized (Sun et al., 2021).

This study demonstrates a proof of concept for bridging the gap by sharing the underlying encoder while keeping the mention scoring and pair scoring modules separate per task (dataset). This ensures that a single model can still perform well on vastly different domains. The motivation is that by sharing the underlying encoder,

the span representations produced could be useful for general coreference resolution, while coreference scoring parameters can learn about specific annotation guidelines or domain-specific text.[8]

Reducing model size

There is little work focused on reducing the model size of a coreference resolution model. Most work focused on efficiency prioritize inference memory usage (Kirstain et al., 2021; Dobrovolskii, 2021) as opposed to number of parameters. These works do not target the encoder. On the other hand, there is a vast literature on compressing encoders for general purpose usage (Sanh et al. (2020a), Jiao et al. (2020), *inter alia*). These models could subsequently be used for finetuning coreference resolution. In this section, I directly target reducing the encoder size using the coreference task objective.

6.3.2 Methods

Section 6.2 suggests that it should be fairly straightforward to apply any of those multitask pruning methods to any task. For this experiment, I use *local movement pruning*, which was one of the competitive approaches previously identified. The previous section also suggests that 1) minimal hyperparameter tuning can yield promising results and 2) multitask models can provide benefits to smaller datasets. I

[8] As mentioned in Section 6.1, assuming a coreference resolution model with N encoder parameters and M span scoring parameters, this would reduce the size of an ensemble of models for k datasets down from $k(N+M)$ to $N+kM$, which may offer benefits over single general model of size $N+M$.

test these claims by adopting similar hyperparameters as Section 6.2 with a minimal search on total training epochs.

Four datasets are used: PreCo (Chen et al., 2018), LitBank (Bamman et al., 2020) which is a collection of literature, QBCoref (Guha et al., 2015) which is a collection of trivial questions, and SARA (Holzenberger and Van Durme, 2021), which is a set of legal statutes (these are also described in Table 2.1).[9] While these four datasets are small, the same splits were usable as training data in Section 5.2.

Concretely, the model encoder is a SpanBERT (Joshi et al., 2020) architecture model which is initialized with the weights of the encoder of a coreference resolution model that has converged on a larger coreference dataset. The weights of the span scoring and linking models are also initialized to the weights of the pretrained model. Mask scores used in local movement pruning are initialized to 0 for all of the weights in the encoder (only). Each training step, an example are sampled from the combined three datasets and the document is encoded with the encoder that is being pruned. The final token representation are used in the ICoref model and those parameters are not pruned.

6.3.3 Proof of concept

One multitask model is pruned to various target densities, and single-task models are also pruned to the same target densities. The models are trained for 40 total

[9]For each of these datasets, only the first split is used. For PreCo, I sample 500/100/500 train, dev, and test documents.

Dataset	Density	Avg. F1			
		PreCo	QBCoref	LitBank	SARA
All	1.0	58.4	64.1	58.1	59.0
	0.9	57.6	63.0	55.2	57.2
	0.7	56.1	60.3	52.1	58.5
All - PreCo	1.0	–	65.6	59.5	67.2
	0.9	–	60.3	49.0	64.0
Shared	1.0	58.8	64.3	59.0	55.7
Shared - Preco	1.0	–	64.4	60.3	68.4
PreCo	1.0	58.8	–	–	–
	0.9	57.0	–	–	–
	0.7	55.7	–	–	–
QBCoref	1.0	–	64.8	–	–
	0.9	–	58.2	–	–
LitBank	1.0	–	–	60.0	–
SARA	1.0	–	–	–	56.6

Table 6.3: Test F1 of several models on each dataset based on whether they are pruned and which dataset(s) they are trained on. All uses all 4 datasets with separate coreference resolution model parameters for each dataset. All - PreCo omits the (larger) PreCo dataset. Shared is a single model shared across all tasks (and Shared - PreCo omits PreCo). PreCo, QBCoref, LitBank, and Sara are single-task models. Pruning any model further resulted in divergence.

epochs, with 4 epochs of initial and final warmup, following similar ratios as Section 6.2 and Sanh et al. (2020b). Models with PreCo were trained for 16 total epochs, with 2 epochs of initial and final warmup. A single maximum span width of 15 was used for all the datasets. The remaining hyperparameters for ICoref are not changed from those in Section 5.2, and the learning rate hyperparameters are not changed from the defaults in Section 6.2. The final model checkpoint is evaluated. Models were trained with pruning rates in the range [0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0 (no pruning)].

Most of the training runs diverged (on the dev. set); the non-diverged model scores

are evaluated on the test sets in Table 6.3. The numerical results show that pruning a multidataset coreference resolution model could be possible, as an encoder was pruned to 70% of the original size. However, substantially more data or training iterations may be required to avoid divergence. Nonetheless, the multitask model did not diverge at 70% for most datasets while the single-task models did. This provides further evidence to support the hypothesis from Section 6.2.5 that auxiliary training tasks could help pruning, or perhaps make convergence smoother. In particular, without PreCo, I found that the model diverged even at 70%.

Looking at the unpruned models, the multitask model appears to be competitive against the single-task models and the baselines. This suggests that it is possible to share an encoder for disparate domains, and reducing a model from $k(N + M)$ parameters to $N + kM$ incurs little cost in accuracy while significantly reducing the number of parameters.

6.3.4 Discussion

Conclusiveness

Since most of the training runs diverged, this speculative experiment suggests that while a movement pruning approach might be viable, it is not necessarily the best option. Nonetheless, without a more exhaustive sweep over all the hyperparameters and comparison to other compression methods, these numerical results are inconclusive.

It does demonstrate that the method described in Section 6.2 may not be as generally applicable out of the box as it was on the tasks used in that study.

This proof of concept shows that it is still possible to apply movement pruning methods to a span-level information extraction task like coreference resolution, which contrasts with the set of tasks studied both by Sanh et al. (2020b) and Section 6.2 that were focused on sentence-level classification or question-answering tasks. In particular, this is one of the first, to my knowledge, attempts to prune the encoder specifically for a coreference resolution model.

Alternatives to compression

One alternative for reducing model size is to use a smaller pretrained encoder, like SpanBERT-base. This was discussed in Section 5.2.4.1. In that setting, the same datasets were used in continued training, and substantially fewer training epochs were needed for the model to converge. Thus, one recommendation based on this proof of concept is that when reducing model size, for smaller datasets, it may be beneficial to start with a smaller encoder altogether rather than attempting to compress larger models. On the other hand, this proof of concept demonstrates that encoder sharing is viable, and so if coreference resolution is needed for multiple, disparate domains, then the total number of parameters can still be reduced by sharing an (unpruned) encoder.

Gordon et al. (2020) suggests that low levels (30%-40%) of pruning should not

affect pre-training loss or downstream transfer. However, that is still does not bridge the gap to encoders that are small to begin with.

Pruning the coreference parameters

This experiment does not prune the coreference resolution parameters. In this case, the size of the encoder (340M) dominates the size of the task-specific parameters. However, the task-specific parameters are not tiny (37M per dataset), and so future work could additionally optimize the size of the model after obtaining span embeddings.

6.4 Future directions

This chapter only explores one potential method for reducing the model size, and more broadly only discusses model compression techniques and one coreference resolution model. However, there are other possible methods for reducing model size. For example, unlike some natural language understanding tasks, coreference resolution has been well-studied for decades and there exist a spectrum from non-neural deterministic models to lightweight learned models. While these perform worse on some datasets compared to their neural counterparts, there is yet to be a comparison across domains or limiting model size, yet some of these approaches may outperform the state of the art models.

More investigation into existing methods may shed light on whether model compression would even be in the right direction. For example, we need to better

understand the types of errors that are made (and corrected) as model size changes. One possibility is that only scaled-up models capture common sense or world knowledge that are essential to resolving a reference, as suggested by work on "emergent abilities" (Wei et al., 2022). And, this knowledge might be retained as the (initially large) model is compressed (Li et al., 2020c). In that case, model compression would obtain the benefits of large language models without incurring the cost of model size. On the other hand, it may be the case that the hardest examples require many parameters; then, practitioners should decide how to trade off accuracy and model size.

If the goal is simply to aggressively reduce model size as a machine learning engineering problem, we already have the toolkits needed, as described in Section 6.1. One could adopt the approach used by Kim and Hassan (2020) and combine all model compression techniques into one system. If there are multiple datasets to target (and there is sufficient data), then the approach described in this chapter can also be included. Nonetheless, doing so gains little insight into what methods work and more crucially, where errors come from and how to fix them.

Stepping back, coreference resolution datasets are highly heterogeneous, as exemplified by the different underlying domains and annotation guidelines. Easier datasets may be resolvable by significantly smaller models than harder ones. Thus, in the setting where there are multiple datasets requiring an encoder to form word or span representations, it's possible that the dataset-specific models should vary significantly in size. Alternatively, at inference, an "early exit" mechanism (Xin et al.,

2020) could be impactful.

Chapter 7

Conclusions and Future Directions

7.1 Summary

In this thesis, I discussed the current state of the field in coreference resolution (Chapter 2). In doing so, I highlighted several limitations of modern neural approaches. These limitations affect efficiency and generalizability of models and hinder their use in real-world applications (Chapter 3). To overcome some of these limitations, I devised: a more time and space efficient model for coreference resolution that maintains competitive accuracy (Chapter 4), a study that shows the effectiveness of using continued training as a method of overcoming data deficiencies (Chapter 5), and a method for reducing the model size of a multitask model (Chapter 6).

Crucially, these methods are aimed to be general purpose: the inference time modifications can, in theory, be made to any publicly released model that follows the E2E framework of detecting-then-linking, as the model parameters are directly lifted from existing models. Meanwhile, the continued training recipe is successful across multiple datasets and languages and the multitask pruning method is a general technique for pruning pretrained language models, which now are an integral component of and ubiquitous in all coreference resolution models.

7.2 Impact

In addition to the studies presented here, I have further used these findings, either directly or in collaboration, to raise concerns around the impracticality of the

OntoNotes-driven development of the field. For example, Yuan et al. (2021) seeks to improve annotation efficiency through active learning and learning through partial annotations, which is achievable through the ICoref model. Toshniwal et al. (2021) is another work which aims to look broadly across many datasets to address the incompatibility of singletons and speaker features.

More efficient memory usage has also been the focus of subsequent work like Kirstain et al. (2021), which reduces the complexity of span representations, and Allaway et al. (2021), which uses a similar incremental algorithm to ICoref for cross-document coreference resolution. Similarly, Schröder et al. (2021) use an incremental model to explore coreference in German books, as books would typically be too long for other models.

LOME (Section 5.1.1) itself is an intermediate result of a substantial subset of work in this thesis. It enjoys the memory benefits of ICoref and makes use of XLM-R as a step towards multilingual generalizability across languages. As a result, the model makes reasonable predictions for inputs of any size in many languages, unlike other memory- or language- bound models.

The code for ICoref is one of few neural coreference resolution codebases that was not designed specifically for OntoNotes. For example, the configuration file structure is designed for users to think about dataset choices, like whether gold mentions should be included and whether singletons should be predicted (or scored). It also encourages users consider the choice of pretrained encoder, checkpoints, and memory usage. With

the code publicly released, I hope that researchers and users of ICoref consider modeling coreference resolution not as a single task and dataset but as a broader challenge with many hills to climb simultaneously.

7.3 Short-term future directions

Looking forward, I suggest a couple challenges that this thesis did not directly address, but are now closer in scope.

Revisiting long-document coreference resolution

Chapter 4 gave a solution to inference memory usage and latency. One of the arguments made in that chapter is that ICoref is able to process and make predictions on longer documents. However, as noted in Table 4.2, the model's accuracy drops as the document size increases. For documents with thousands of tokens, perhaps the simplification of clusters to single vectors adopted by limited memory incremental models is incorrect.

Instead, maybe the correct formulation is to index the entity representations with richer information, like predicted types relations to other entities. In other words, this would grow a knowledge *graph* for a document rather than an entity *set*. There could be fairly minor modifications to model specific relations between entities already collected by the entity set in the ICoref model, and this would be an interesting

incremental direction to explore. This model, like ICoref would run locally (over a segment) and periodically link up entities in its set to the knowledge "graph," and subsequently update the graph. This is different from the current formulation, which assumes that the contextual information and span embeddings contain sufficient information to represent an entity. Extrapolating this extension further to multiple documents and more complex information in the graph, this converges towards the full SM-KBP task (Section 3.2.1).

However, this may not be the right direction. Instead, perhaps our current methods are still best suited for shorter passages of text and local coreference resolution. For longer documents or passages, there may need to be another layer which integrates the predictions made by local models. This way, local models could be optimized for shorter windows while a separate model, trained differently, aggregates across long distances. Such a system may have applications to cross-document coreference resolution, or perhaps unite the cross-document and within-document subfields of the field.

Multilinguality and zero anaphora

In Chapter 5, I discussed methods for creating multilingual and cross-lingual models. One limitation of this approach is its disregard towards any language-specific coreference phenomena. For example, consider the following Chinese sentence from Chapter 21 of Jurafsky and Martin (2021):

- [我] 前一会精神上太紧张。[0] 现在比较平静了。
  [I] was too nervous a while ago. ... [0] am now calmer.

In this example, the subject (I) in the second sentence is omitted. As discussed in Section 2.2.1, this is present in several languages and even in certain English domains. Subsequent directions in the multilingual and generalization direction would be to devise a more flexible scheme for mention representation *and evaluation* to accommodate features of other languages that imply referents, like zero anaphora, traces, morphology, and split mentions.

Dialogue and social coreference resolution

Another area of focus is on multi-party communications, which is a general domain present in everyday speech that is challenging for coreference resolution and also underexplored. While several of the datasets in Section 2.2.1 are in the dialogue domain, they are almost all between two people or task-driven. The errors and ambiguities in a multi-person conversation and in social settings (e.g. Figure 4.3) are different from the two-person datasets.

One reason for the underexploration is the lack of datasets. Given a dataset, there are many more questions we can ask just beyond what was asked in this thesis. For example, social situations carry common ground, background context, and possibly even visual cues: how can those be used to improve disambiguation? Multi-person conversations often make use of plural entities with overlapping or changing subsets

of the speakers. How can those be discerned? Additionally, can we infer who the listeners are for each utterance, and does that help resolve references?

In particular, multi-party communications are real-time and additional data would complement the online modeling from Section 4.3. Recently, we have started to work on addressing the paucity of datasets for multi-party communication, and in a multilingual context (Zheng et al., 2022).

Bridging, deixis, and visual coreference

This work does not interact with most of the work in discourse on bridging (e.g. "Washington" to refer to the U.S. Government), discourse deixis (e.g. "this" and "that" in discourse (Webber, 1988)), and visual coreference. The ICoref model is amenable to extensions to some extent: for example, bridging can be predicted using a different type of pairwise scorer, where the entity cluster representation is only updated partially or not at all. Discourse deixis is harder, as it may require encoding longer spans or discourse units as markables; still, the ICoref model has an advantage in that it will not be as memory-limited compared to other models due to the longer spans. Finally, the model is not extendable to visual coreference, when defined between images and captions. However, an incremental and online model might be amenable to visual coreference in video, although this would need further exploration.

Comparison of annotation interfaces

It is evident, based on Chapter 5, that in-domain labels are always beneficial. Therefore, to build a model for a new domain or language, data needs to be annotated. While there is already some work in active learning (Li et al., 2020b; Yuan et al., 2021) for selecting which spans to annotate, there are other parts of the data collection pipeline that have not been addressed. What is the fastest way to collect and clean annotations for the new data? Which ones are the easiest for a non-expert to use? What are the best practices for displaying instructions, selecting spans, and adjudicating annotations?

Unlike other tasks in NLP, collecting annotations via crowdsourcing is nascent for coreference resolution. However, we can't rely on the datasets in Section 2.2.1 for all applications, and so these open questions still need answers.

Automatic data annotation

Devlin et al. (2019) (among others) showed that by using a self-supervised training objective, models can leverage magnitudes of unannotated data. Coreference resolution has intersected with this idea by using Wikipedia hyperlinks (Kocijan et al., 2019) and through the use of silver coreference data (Ye et al., 2020). However, there has yet to be a compelling method for a self-supervised coreference objective that can be used on arbitrary collections on unlabeled data.

Again, Chapter 5 showed that to transfer to new domains, we need annotations for a particular domain. Furthermore, success of pretrained language models on diagnostic tasks suggest that large language models can be good coreference resolvers (Beyer et al., 2021). One path forward in this direction could be to leverage the knowledge within large language models to label a few documents of an arbitrary dataset and subsequently (continued) train a smaller model using these documents. This could be a recipe for achieving competitive, dataset-adapted baselines for any dataset.

7.4 Rethinking the "task" of coreference

There are also longer-term goals for coreference resolution to which this work is only one of many steps. One of the main challenges is to properly define the scope and task description of coreference resolution. Even in this thesis, I presented two versions: one with and one without given mention boundaries. In this section, I suggest that we should no longer think of coreference resolution as a single task or even a useful standalone research objective.

For many years, coreference linking with system (gold) mentions was a setting that researchers evaluated on (e.g. Recasens et al. (2010)). In fact, I argued in Chapter 3 that this setting *was* still useful because of the gap in performance we observed on datasets other than OntoNotes and *because OntoNotes itself is incomplete*. In other words, I believe the community acted too hastily in throwing away a useful

experimental and evaluation setting. However, now that we do finally have models that explicitly predict singletons (Section 2.5.2, Section 4.2) and more widespread evaluation on multiple datasets, we might be able to ignore the gold mentions setting for good, at least in research (in full systems like LOME (Section 3.2), we would still find it useful). To support this, consider the trends in Figure 5.3 and Figure 5.4. We see that the end-to-end task performance roughly matches the behavior of mention detection and coreference linking, as does the rough relative comparisons between models and data quantity. So methods that work well on one interpretation of the task may work well more generally and vice versa.

Next, coreference resolution is rarely useful as a standalone task. Instead, its outputs are used by downstream systems or analysts for a different end goal. In those scenarios, it would be more fruitful to co-develop the coreference aspects jointly with the rest of the system. In particular, I believe we are close to the point where a "good enough" initial model should be readily and cheaply available and that model development should no longer be a priority. Many of the recent performance gains are due to general NLP advances, and coreference resolution will continue to benefit from those innovations.[1] Meanwhile, using one of these models as a starting point is cheap and this thesis presented only a small number of ways of extending those models based on real-world constraints like compute and data. Thus, the future direction in coreference resolution should be studying how it can be best integrated as a component

[1]Core modeling contributions might improve performance, but it's difficult to know *a priori* how many of those contributions would simply be subsumed by improved general NLP and NLU methods.

of a larger system.

As a historical point of comparison, consider Penn Treebank parsing. It is now a less common topic of research but still a commonly used tool taken for granted that is used by other systems. It also enjoyed benefits from improved word representations. Now, sufficiently well-performing and performant toolkits are readily available for any practitioner. Of course, syntactic parsing is not solved, especially in low-resource languages or less-studied domains. Nonetheless, the technology is usually good enough to be used without much scrutiny by downstream applications as a first pass. Similarly, I believe coreference resolution will reached that status in the near future, if we aren't already there.

Implicitly defined clusters

Concretely, here is one suggestion of "tight" integration in a real-world task. Suppose a task requires resolving coreferences. The proposed approach from this thesis is to take the approach from Section 3.2 and create a dedicated model and collect sufficient data to properly (continued) train that model as described in Section 5.2.

Instead, I suggest a different approach, looking towards future models and joint or "deeper" end-to-end systems. It might be sufficient to inject the bias that latent entity clusters, or even just antecedent links, should exist at all. In doing so, they could be implicitly used and helpful towards a downstream task without explicit coreference supervision. There is some precedent to these types of approaches, e.g. Kim et al.

(2019) assumes latent trees for syntactic parsing.

One full-stack model

As another example, consider the role of coreference resolution in an information extraction stack (i.e. the SM-KBP task). The long-term vision would be to create a single end-to-end neural model that completes the full stack quickly and efficiently on arbitrarily large collections of documents with high accuracy. Drawing from the themes of this thesis, this would only be possible with efficient components throughout the model and with task-specific modules each trained on different tasks (or datasets).

I claim that eventually, these model should not have an explicit E2E-based model as an underlying component. This is because the task data may not have exhaustive annotation and these models are increasingly driven large language models shared across all components. Thus, it is unclear how research solely on coreference resolution models could contribute to such a model.

Regardless of what the model looks like, it would imply that coreference resolution was successfully integrated into a single, end-to-end, information extraction model, which would conclude years of research in joint modeling (e.g. Durrett and Klein (2014)). However, this may require a new approach independent from the lessons learned from this thesis. Thus, I suggest future-looking research to first identify *why* we are interested in coreference resolution (what precisely is the downstream task? what data exists?) before studying it.

7.5 Closing Remarks

This thesis advances the current state of efficient models for coreference resolution through a focus on inference efficiency, use of data transfer, and model compression. However, it is only the start of efficiently scaling up coreference resolution. There are still a multitude of questions around document length, non-English coreference, nonconversational text like dialogue, looser notions of coreference like bridging, and annotation. Research in each of these areas could be enabled by some of the lessons learned from this thesis.

Stepping back, however, I envision coreference resolution will always be used as a component of a real-world system for a different task. The requirements for each task will be different, and so it is important to explore *general* approaches for data annotation and model development. At the same time, coreference could be a deeply embedded part of those systems and so each system would require customized methodology for handling coreference. Perhaps the biggest open problem is creating methods for coreference resolution that scale across a variety of downstream needs when each uses a different definition of coreference.

Bibliography

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta (Nov. 2021). "Muppet: Massive Multi-task Representations with Pre-Finetuning". *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pages 5799–5811. DOI: 10.18653/v1/2021.emnlp-main.468. URL: https://aclanthology.org/2021.emnlp-main.468 (cited on page 147).

Alan Akbik and Yunyao Li (Aug. 2016). "POLYGLOT: Multilingual Semantic Role Labeling with Unified Labels". *Proceedings of ACL-2016 System Demonstrations*. Berlin, Germany: Association for Computational Linguistics, pages 1–6. DOI: 10.18653/v1/P16-4001. URL: https://aclanthology.org/P16-4001 (cited on page 112).

Berfin Aktaş, Veronika Solopova, Annalena Kohnert, and Manfred Stede (Nov. 2020). "Adapting Coreference Resolution to Twitter Conversations". *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for

Computational Linguistics, pages 2454–2460. DOI: 10.18653/v1/2020.findings-emnlp.222. URL: https://aclanthology.org/2020.findings-emnlp.222 (cited on pages 119, 121).

Emily Allaway, Shuai Wang, and Miguel Ballesteros (Nov. 2021). "Sequential Cross-Document Coreference Resolution". *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pages 4659–4671. DOI: 10.18653/v1/2021.emnlp-main.382. URL: https://aclanthology.org/2021.emnlp-main.382 (cited on pages 44, 171).

Abdulrahman Aloraini, Juntao Yu, and Massimo Poesio (Dec. 2020). "Neural Coreference Resolution for Arabic". *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*. Barcelona, Spain (online): Association for Computational Linguistics, pages 99–110. URL: https://aclanthology.org/2020.crac-1.11 (cited on pages 133, 138).

Jose M. Alvarez and Mathieu Salzmann (2016). "Learning the Number of Neurons in Deep Networks". *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. Barcelona, Spain: Curran Associates Inc., 2270–2278. ISBN: 9781510838819 (cited on page 143).

Giuseppe Attardi, Maria Simi, and Stefano Dei Rossi (July 2010). "TANL-1: Coreference Resolution by Parse Analysis and Similarity Clustering". *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden:

Association for Computational Linguistics, pages 108–111. URL: https://aclanthology.org/S10-1022 (cited on page 133).

Amit Bagga and Breck Baldwin (Aug. 1998). "Entity-Based Cross-Document Coreferencing Using the Vector Space Model". *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*. Montreal, Quebec, Canada: Association for Computational Linguistics, pages 79–85. DOI: 10.3115/980845.980859. URL: https://aclanthology.org/P98-1012 (cited on pages 20, 77).

Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jin Jin, Xin Jiang, Qun Liu, Michael Lyu, and Irwin King (Aug. 2021). "BinaryBERT: Pushing the Limit of BERT Quantization". *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pages 4334–4348. DOI: 10.18653/v1/2021.acl-long.334. URL: https://aclanthology.org/2021.acl-long.334 (cited on page 144).

Collin F. Baker, Charles J. Fillmore, and John B. Lowe (1998). "The Berkeley FrameNet Project". *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*. URL: https://aclanthology.org/C98-1013 (cited on page 110).

David Bamman, Olivia Lewke, and Anya Mansoor (May 2020). "An Annotated Dataset of Coreference in English Literature". English. *Proceedings of the 12th Language*

Resources and Evaluation Conference. Marseille, France: European Language Resources Association, pages 44–54. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.6 (cited on pages 9, 12, 95, 119, 124, 125, 162).

Iz Beltagy, Matthew E. Peters, and Arman Cohan (2020). *Longformer: The Long-Document Transformer*. DOI: 10.48550/ARXIV.2004.05150. URL: https://arxiv.org/abs/2004.05150 (cited on page 34).

Matan Ben Noach and Yoav Goldberg (Dec. 2020). "Compressing Pre-trained Language Models by Matrix Decomposition". *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, pages 884–889. URL: https://aclanthology.org/2020.aacl-main.88 (cited on pages 144, 157, 158).

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel (May 2022). "BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models". *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Dublin, Ireland: Association for Computational Linguistics, pages 1–9. DOI: 10.18653/v1/2022.acl-short.1. URL: https://aclanthology.org/2022.acl-short.1 (cited on page 160).

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville (2013). "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation". *ArXiv* abs/1308.3432 (cited on page 151).

Eric Bengtson and Dan Roth (Oct. 2008). "Understanding the Value of Features for Coreference Resolution". *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: Association for Computational Linguistics, pages 294–303. URL: https://aclanthology.org/D08-1031 (cited on pages 26, 38, 39).

Anne Beyer, Sharid Loáiciga, and David Schlangen (June 2021). "Is Incoherence Surprising? Targeted Evaluation of Coherence Prediction from Language Models". *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pages 4164–4173. DOI: 10.18653/v1/2021.naacl-main.328. URL: https://aclanthology.org/2021.naacl-main.328 (cited on page 177).

Anders Björkelund and Jonas Kuhn (June 2014). "Learning Structured Perceptrons for Coreference Resolution with Latent Antecedents and Non-local Features". *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pages 47–57. DOI: 10.3115/v1/P14-1005. URL: https://aclanthology.org/P14-1005 (cited on page 25).

Elizabeth A Boyle, Anne H Anderson, and Alison Newlands (1994). "The effects of visibility on dialogue and performance in a cooperative problem solving task". *Language and speech* 37.1, pages 1–20 (cited on page 91).

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei (2020a). "Language Models are Few-Shot Learners". *ArXiv* abs/2005.14165 (cited on page 43).

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei (2020b). "Language Models are Few-Shot Learners". arXiv: 2005.14165 [cs.CL] (cited on pages 119, 130).

Marc Brysbaert (2019). "How many words do we read per minute? A review and meta-analysis of reading rate". *Journal of memory and language* 109, page 104047 (cited on page 104).

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil (2006). "Model compression". *KDD '06* (cited on page 143).

A. E. Budnikov, S Yu Toldova, D. S. Zvereva, D. M. Maximova, and M. I. Ionov (2019). "RU-EVAL-2019: Evaluating Anaphora and Coreference Resolution For Russian". *Computational Linguistics and Intellectual Technologies - Supplementary Volume*. URL: http://www.dialog-21.ru/en/evaluation/2019/disambiguation/anaphora/ (cited on pages 14, 114).

Michael Bugert, Nils Reimers, and Iryna Gurevych (Nov. 2021). "Generalizing Cross-Document Event Coreference Resolution Across Multiple Corpora". *Computational Linguistics* 47.3, pages 575–614. DOI: 10.1162/coli_a_00407. URL: https://aclanthology.org/2021.cl-3.18 (cited on page 44).

Avi Caciularu, Arman Cohan, Iz Beltagy, Matthew Peters, Arie Cattan, and Ido Dagan (Nov. 2021). "CDLM: Cross-Document Language Modeling". *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pages 2648–2662. DOI: 10.18653/v1/2021.findings-emnlp.225. URL: https://aclanthology.org/2021.findings-emnlp.225 (cited on page 92).

Jie Cai and Michael Strube (Sept. 2010). "Evaluation Metrics For End-to-End Coreference Resolution Systems". *Proceedings of the SIGDIAL 2010 Conference*. Tokyo, Japan: Association for Computational Linguistics, pages 28–36. URL: https://aclanthology.org/W10-4305 (cited on page 20).

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan (Aug. 2021a). "Cross-document Coreference Resolution over Predicted Mentions".

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. Online: Association for Computational Linguistics, pages 5100–5107. DOI: 10.18653/v1/2021.findings-acl.453. URL: https://aclanthology.org/2021.findings-acl.453 (cited on page 44).

Arie Cattan, Sophie Johnson, Daniel S Weld, Ido Dagan, Iz Beltagy, Doug Downey, and Tom Hope (2021b). "SciCo: Hierarchical Cross-Document Coreference for Scientific Concepts". *3rd Conference on Automated Knowledge Base Construction*. URL: https://openreview.net/forum?id=OFLbgUP04nC (cited on page 16).

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia (Aug. 2017). "SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation". *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, pages 1–14. DOI: 10.18653/v1/S17-2001. URL: https://aclanthology.org/S17-2001 (cited on page 158).

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos (Nov. 2020). "LEGAL-BERT: The Muppets straight out of Law School". *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pages 2898–2904. DOI: 10.18653/v1/2020.findings-emnlp.261. URL: https://aclanthology.org/2020.findings-emnlp.261 (cited on page 137).

Chen Chen and Vincent Ng (Dec. 2012). "Chinese Noun Phrase Coreference Resolution: Insights into the State of the Art". *Proceedings of COLING 2012: Posters*. Mumbai, India: The COLING 2012 Organizing Committee, pages 185–194. URL: https://aclanthology.org/C12-2019 (cited on page 133).

Chen Chen and Vincent Ng (Oct. 2013). "Chinese Zero Pronoun Resolution: Some Recent Advances". *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pages 1360–1365. URL: https://aclanthology.org/D13-1135 (cited on page 11).

Hong Chen, Zhenhua Fan, Hao Lu, Alan Yuille, and Shu Rong (2018). "PreCo: A Large-scale Dataset in Preschool Vocabulary for Coreference Resolution". *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pages 172–181. DOI: 10.18653/v1/D18-1016. URL: https://aclanthology.org/D18-1016 (cited on pages 9, 12, 17, 59, 123, 162).

Yu-Hsin Chen and Jinho D. Choi (Sept. 2016). "Character Identification on Multiparty Conversation: Identifying Mentions of Characters in TV Shows". *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Los Angeles: Association for Computational Linguistics, pages 90–100. DOI: 10.18653/v1/W16-3612. URL: https://aclanthology.org/W16-3612 (cited on page 12).

Yunmo Chen, Seth Ebner, Tongfei Chen, Patrick Xia, Elias Stengel-Eskin, Tzu-Ray Su, J. Edward Hu, Nils Holzenberger, Ryan Culkin, Craig Harman, Max Thomas, Thomas Lippincott, Aaron Steven White, Kyle Rawlins, and Benjamin Van Durme (2019). "NIST TAC SM-KBP 2019 System Description: JHU/UR Framework". *TAC* (cited on pages 52, 62).

Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang (2017). "A Survey of Model Compression and Acceleration for Deep Neural Networks". *ArXiv* abs/1710.09282 (cited on page 142).

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever (2019). "Generating Long Sequences with Sparse Transformers". *URL https://openai.com/blog/sparse-transformers* (cited on page 80).

Jinho D. Choi and Henry Y. Chen (June 2018). "SemEval 2018 Task 4: Character Identification on Multiparty Dialogues". *Proceedings of The 12th International Workshop on Semantic Evaluation*. New Orleans, Louisiana: Association for Computational Linguistics, pages 57–64. DOI: 10.18653/v1/S18-1007. URL: https://aclanthology.org/S18-1007 (cited on page 12).

Christopher Cieri, James Fiumara, Stephanie Strassel, Jonathan Wright, Denise DiPersio, and Mark Liberman (May 2020). "A Progress Report on Activities at the Linguistic Data Consortium Benefitting the LREC Community". English. *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pages 3449–3456. ISBN:

979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.423 (cited on page 113).

Jonathan H Clark and José P González-Brenes (2008). "Coreference resolution: Current trends and future directions". *Language and Statistics II Literature Review* 14 (cited on page 24).

Kevin Clark and Christopher D. Manning (Nov. 2016a). "Deep Reinforcement Learning for Mention-Ranking Coreference Models". *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pages 2256–2262. DOI: 10.18653/v1/D16-1245. URL: https://aclanthology.org/D16-1245 (cited on pages 49, 70, 118).

Kevin Clark and Christopher D. Manning (Aug. 2016b). "Improving Coreference Resolution by Learning Entity-Level Distributed Representations". *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pages 643–653. DOI: 10.18653/v1/P16-1061. URL: https://aclanthology.org/P16-1061 (cited on pages 25, 41, 49, 70, 118, 160).

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov (July 2020). "Unsupervised Cross-lingual Representation Learning at Scale". *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics,

pages 8440–8451. DOI: 10.18653/v1/2020.acl-main.747. URL: https://aclanthology.org/2020.acl-main.747 (cited on pages 63, 110, 112, 127, 134, 137).

Agata Cybulska and Piek Vossen (May 2014). "Using a sledgehammer to crack a nut? Lexical diversity and event coreference resolution". *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), pages 4545–4552. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/840_Paper.pdf (cited on page 16).

Ido Dagan, Oren Glickman, and Bernardo Magnini (2005). "The PASCAL recognising textual entailment challenge". *Machine Learning Challenges Workshop*. Springer, pages 177–190 (cited on page 158).

Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie (2020). "Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey". *Proceedings of the IEEE* 108.4, pages 485–532. DOI: 10.1109/JPROC.2020.2976475 (cited on page 142).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (June 2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". *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)*. Minneapolis, Minnesota: Association for

Computational Linguistics, pages 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423 (cited on pages 47, 63, 146, 152, 176).

Vladimir Dobrovolskii (Nov. 2021). "Word-Level Coreference Resolution". *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pages 7670–7675. DOI: 10.18653/v1/2021.emnlp-main.605. URL: https://aclanthology.org/2021.emnlp-main.605 (cited on pages 39, 41, 161).

William B Dolan and Chris Brockett (2005). "Automatically constructing a corpus of sentential paraphrases". *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)* (cited on page 158).

Greg Durrett, David Hall, and Dan Klein (Aug. 2013). "Decentralized Entity-Level Modeling for Coreference Resolution". *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pages 114–124. URL: https://aclanthology.org/P13-1012 (cited on page 24).

Greg Durrett and Dan Klein (Oct. 2013). "Easy Victories and Uphill Battles in Coreference Resolution". *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pages 1971–1982. URL: https://aclanthology.org/D13-1203 (cited on pages 48, 50, 133).

Greg Durrett and Dan Klein (2014). "A Joint Model for Entity Analysis: Coreference, Typing, and Linking". *Transactions of the Association for Computational Linguistics* 2, pages 477–490. DOI: 10.1162/tacl_a_00197. URL: https://aclanthology.org/Q14-1037 (cited on pages 39, 180).

Pradheep Elango (2006). "Coreference Resolution : A Survey" (cited on page 24).

Micha Elsner and Eugene Charniak (June 2008). "Coreference-inspired Coherence Modeling". *Proceedings of ACL-08: HLT, Short Papers*. Columbus, Ohio: Association for Computational Linguistics, pages 41–44. URL: https://aclanthology.org/P08-2011 (cited on page 3).

Angela Fan, Edouard Grave, and Armand Joulin (2019). "Reducing Transformer Depth on Demand with Structured Dropout". *arXiv preprint arXiv:1909.11556* (cited on page 143).

Chun Fan, Jiwei Li, Xiang Ao, Fei Wu, Yuxian Meng, and Xiaofei Sun (2021). *Layer-wise Model Pruning based on Mutual Information*. arXiv: 2108.12594 [cs.CL] (cited on page 149).

Xing Fan, Emilio Monti, Lambert Mathias, and Markus Dreyer (Aug. 2017). "Transfer Learning for Neural Semantic Parsing". *Proceedings of the 2nd Workshop on Representation Learning for NLP*. Vancouver, Canada: Association for Computational Linguistics, pages 48–56. DOI: 10.18653/v1/W17-2607. URL: https://aclanthology.org/W17-2607 (cited on page 122).

Francis Ferraro, Max Thomas, Matthew R. Gormley, Travis Wolfe, Craig Harman, and Benjamin Van Durme (2014). "Concretely Annotated Corpora". *4th Workshop on Automated Knowledge Base Construction (AKBC)*. URL: http://www.akbc.ws/2014/submissions/akbc2014_submission_18.pdf (cited on page 112).

Jonathan Frankle and Michael Carbin (2019). "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks". *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. URL: https://openreview.net/forum?id=rJl-b3RcF7 (cited on page 143).

Trevor Gale, Matei Zaharia, Cliff Young, and Erich Elsen (2020). "Sparse GPU Kernels for Deep Learning". *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. SC '20. Atlanta, Georgia: IEEE Press. ISBN: 9781728199986 (cited on pages 148, 154).

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer (July 2018). "AllenNLP: A Deep Semantic Natural Language Processing Platform". *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*. Melbourne, Australia: Association for Computational Linguistics, pages 1–6. DOI: 10.18653/v1/W18-2501. URL: https://aclanthology.org/W18-2501 (cited on page 120).

Abbas Ghaddar and Phillippe Langlais (May 2016). "WikiCoref: An English Coreference-annotated Corpus of Wikipedia Articles". *Proceedings of the Tenth*

International Conference on Language Resources and Evaluation (LREC'16). Portorož, Slovenia: European Language Resources Association (ELRA), pages 136–142. URL: https://aclanthology.org/L16-1021 (cited on page 12).

Mitchell Gordon, Kevin Duh, and Nicholas Andrews (July 2020). "Compressing BERT: Studying the Effects of Weight Pruning on Transfer Learning". *Proceedings of the 5th Workshop on Representation Learning for NLP*. Online: Association for Computational Linguistics, pages 143–155. DOI: 10.18653/v1/2020.repl4nlp-1.18. URL: https://aclanthology.org/2020.repl4nlp-1.18 (cited on page 165).

Artem M. Grachev, Dmitry I. Ignatov, and Andrey V. Savchenko (2017). "Neural Networks Compression for Language Modeling". *Pattern Recognition and Machine Intelligence*. Edited by B. Uma Shankar, Kuntal Ghosh, Deba Prasad Mandal, Shubhra Sankar Ray, David Zhang, and Sankar K. Pal. Cham: Springer International Publishing, pages 351–357. ISBN: 978-3-319-69900-4 (cited on page 144).

Ralph Grishman and Beth Sundheim (1996). "Message Understanding Conference- 6: A Brief History". *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*. URL: https://aclanthology.org/C96-1079 (cited on page 2).

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li (Apr. 2017). "Learning to Translate in Real-time with Neural Machine Translation". *Proceedings of the 15th Conference of the European Chapter of the Association for*

Computational Linguistics: Volume 1, Long Papers. Valencia, Spain: Association for Computational Linguistics, pages 1053–1062. URL: https://aclanthology.org/E17-1099 (cited on page 94).

Anupam Guha, Mohit Iyyer, Danny Bouman, and Jordan Boyd-Graber (2015). "Removing the Training Wheels: A Coreference Dataset that Entertains Humans and Challenges Computers". *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, pages 1108–1118. DOI: 10.3115/v1/N15-1117. URL: https://aclanthology.org/N15-1117 (cited on pages 12, 95, 124, 133, 162).

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith (July 2020). "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks". *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pages 8342–8360. DOI: 10.18653/v1/2020.acl-main.740. URL: https://aclanthology.org/2020.acl-main.740 (cited on pages 105, 122, 129).

Benjamin Haeffele, Eric Young, and Rene Vidal (2014). "Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing". *Proceedings of the 31st International Conference on Machine Learning*. Edited by Eric P. Xing and Tony Jebara. Volume 32. Proceedings of Machine Learning

Research 2. Bejing, China: PMLR, pages 2007–2015. URL: https://proceedings.mlr.press/v32/haeffele14.html (cited on page 144).

Aria Haghighi and Dan Klein (Aug. 2009). "Simple Coreference Resolution with Rich Syntactic and Semantic Features". *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, pages 1152–1161. URL: https://aclanthology.org/D09-1120 (cited on page 25).

Aria Haghighi and Dan Klein (June 2010). "Coreference Resolution in a Modular, Entity-Centered Model". *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California: Association for Computational Linguistics, pages 385–393. URL: https://aclanthology.org/N10-1061 (cited on page 54).

Song Han, Jeff Pool, John Tran, and William J. Dally (2015). "Learning Both Weights and Connections for Efficient Neural Networks". *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'15. Montreal, Canada: MIT Press, 1135–1143 (cited on pages 143, 150).

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer (July 2018). "Jointly Predicting Predicates and Arguments in Neural Semantic Role Labeling". *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for

Computational Linguistics, pages 364–369. DOI: 10.18653/v1/P18-2058. URL: https://aclanthology.org/P18-2058 (cited on page 51).

Erhard W. Hinrichs, Sandra Kübler, and Karin Naumann (June 2005). "A Unified Representation for Morphological, Syntactic, Semantic, and Referential Annotations". *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*. Ann Arbor, Michigan: Association for Computational Linguistics, pages 13–20. URL: https://aclanthology.org/W05-0303 (cited on page 14).

G. E. Hinton and R. R. Salakhutdinov (2006). "Reducing the Dimensionality of Data with Neural Networks". *Science* 313.5786, pages 504–507. ISSN: 0036-8075. DOI: 10.1126/science.1127647. eprint: http://science.sciencemag.org/content/313/5786/504.full.pdf. URL: http://science.sciencemag.org/content/313/5786/504 (cited on page 119).

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean (2015). "Distilling the Knowledge in a Neural Network". *ArXiv* abs/1503.02531 (cited on page 143).

Jerry R. Hobbs (1978). "Resolving pronoun references". *Lingua* 44.4, pages 311–338. ISSN: 0024-3841. DOI: https://doi.org/10.1016/0024-3841(78)90006-2. URL: https://www.sciencedirect.com/science/article/pii/0024384178900062 (cited on page 23).

Nils Holzenberger and Benjamin Van Durme (Aug. 2021). "Factoring Statutory Reasoning as Language Understanding Challenges". *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International*

Joint Conference on Natural Language Processing (Volume 1: Long Papers). Online: Association for Computational Linguistics, pages 2742–2758. DOI: 10.18653/v1/2021.acl-long.213. URL: https://aclanthology.org/2021.acl-long.213 (cited on pages 12, 125, 133, 162).

Véronique Hoste and Guy De Pauw (May 2006). "KNACK-2002: a Richly Annotated Corpus of Dutch Written Text". *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. Genoa, Italy: European Language Resources Association (ELRA). URL: http://www.lrec-conf.org/proceedings/lrec2006/pdf/342_pdf.pdf (cited on page 14).

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly (2019). "Parameter-Efficient Transfer Learning for NLP". *Proceedings of the 36th International Conference on Machine Learning*. Edited by Kamalika Chaudhuri and Ruslan Salakhutdinov. Volume 97. Proceedings of Machine Learning Research. PMLR, pages 2790–2799. URL: https://proceedings.mlr.press/v97/houlsby19a.html (cited on pages 146, 160).

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel (June 2006). "OntoNotes: The 90% Solution". *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. New York City, USA: Association for Computational Linguistics, pages 57–60. URL: https://aclanthology.org/N06-2015 (cited on page 9).

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen (2021). "LoRA: Low-Rank Adaptation of Large Language Models". *ArXiv* abs/2106.09685 (cited on page 146).

Yerlan Idelbayev and Miguel A. Carreira-Perpinan (2020). "Low-Rank Compression of Neural Nets: Learning the Rank of Each Layer". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (cited on page 144).

Ryu Iida and Massimo Poesio (June 2011). "A Cross-Lingual ILP Solution to Zero Anaphora Resolution". *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, pages 804–813. URL: https://aclanthology.org/P11-1081 (cited on page 11).

Shankar Iyer, Nikhil Dandekar, and Kornel Csernai (2017). *First Quora Dataset Release: Question Pairs*. URL: https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs (visited on 09/22/2021) (cited on page 152).

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew G. Howard, Hartwig Adam, and Dmitry Kalenichenko (2018). "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference". *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2704–2713 (cited on page 144).

Fan Jiang and Trevor Cohn (June 2021). "Incorporating Syntax and Semantics in Coreference Resolution with Heterogeneous Graph Attention Network". *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pages 1584–1591. DOI: 10.18653/v1/2021.naacl-main.125. URL: https://aclanthology.org/2021.naacl-main.125 (cited on pages 39, 41, 42, 53).

Fan Jiang and Trevor Cohn (2022). "Incorporating Constituent Syntax for Coreference Resolution". *ArXiv* abs/2202.10710 (cited on pages 39, 41).

Yiwei Jiang, Klim Zaporojets, Johannes Deleu, Thomas Demeester, and Chris Develder (Dec. 2020). "Recipe Instruction Semantics Corpus (RISeC): Resolving Semantic Structure and Zero Anaphora in Recipes". *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, pages 821–826. URL: https://aclanthology.org/2020.aacl-main.82 (cited on page 14).

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu (Nov. 2020). "TinyBERT: Distilling BERT for Natural Language Understanding". *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pages 4163–4174.

DOI: 10.18653/v1/2020.findings-emnlp.372. URL: https://aclanthology.org/2020.findings-emnlp.372 (cited on pages 152, 161).

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, pages 64–77. DOI: 10.1162/tacl_a_00300. URL: https://aclanthology.org/2020.tacl-1.5 (cited on pages 34, 40, 41, 70–73, 75–78, 85, 114, 118, 122, 127, 130, 162).

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld (Nov. 2019). "BERT for Coreference Resolution: Baselines and Analysis". *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pages 5803–5808. DOI: 10.18653/v1/D19-1588. URL: https://aclanthology.org/D19-1588 (cited on pages 23, 34, 41, 44, 70–73, 75, 76, 78, 81, 83, 84, 96, 118, 122, 125, 127).

Vidur Joshi, Matthew Peters, and Mark Hopkins (July 2018). "Extending a Parser to Distant Domains Using a Few Dozen Partially Annotated Examples". *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pages 1190–1199. DOI: 10.18653/v1/P18-1110. URL: https://aclanthology.org/P18-1110 (cited on pages 122, 130).

Norman P. Jouppi, Cliff Young, Nishant Patil, David A. Patterson, Gaurav Agrawal, Raminder Singh Bajwa, Sarah Bates, Suresh Bhatia, Nanette J. Boden, Al Borchers, Rick Boyle, Pierre luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert B. Hagmann, C. Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Daniel Hurt, Julian Ibarz, Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Daniel Killebrew, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle A. Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad Samadiani, Chris Severn, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Bo Tian, Horia Toma, Erick Tuttle, Vijay Vasudevan, Richard Walter, Walter Wang, Eric Wilcox, and Doe Hyun Yoon (2017). "In-datacenter performance analysis of a tensor processing unit". *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, pages 1–12 (cited on page 3).

Dan Jurafsky and James H Martin (2021). *Speech and Language Processing (3rd ed draft)* (cited on pages 2, 173).

Ben Kantor and Amir Globerson (July 2019). "Coreference Resolution with Entity Equalization". *Proceedings of the 57th Annual Meeting of the Association*

for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, pages 673–677. DOI: 10.18653/v1/P19-1066. URL: https://aclanthology.org/P19-1066 (cited on pages 33, 41, 70).

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson (Aug. 2021a). "Parameter-efficient Multi-task Fine-tuning for Transformers via Shared Hypernetworks". *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pages 565–576. DOI: 10.18653/v1/2021.acl-long.47. URL: https://aclanthology.org/2021.acl-long.47 (cited on page 146).

Rabeeh Karimi Mahabadi, Sebastian Ruder, and James Henderson (2021b). *Compacter: Efficient Low-Rank Hypercomplex Adapter Layers*. arXiv: 2106.04647 [cs.CL] (cited on page 146).

Huda Khayrallah, Brian Thompson, Kevin Duh, and Philipp Koehn (July 2018). "Regularized Training Objective for Continued Training for Domain Adaptation in Neural Machine Translation". *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. Melbourne, Australia: Association for Computational Linguistics, pages 36–44. DOI: 10.18653/v1/W18-2705. URL: https://aclanthology.org/W18-2705 (cited on pages 119, 122).

Sopan Khosla, Juntao Yu, Ramesh Manuvinakurike, Vincent Ng, Massimo Poesio, Michael Strube, and Carolyn Rosé (Nov. 2021). "The CODI-CRAC 2021 Shared

Task on Anaphora, Bridging, and Discourse Deixis in Dialogue". *Proceedings of the CODI-CRAC 2021 Shared Task on Anaphora, Bridging, and Discourse Deixis in Dialogue*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pages 1–15. DOI: 10.18653/v1/2021.codi-sharedtask.1. URL: https://aclanthology.org/2021.codi-sharedtask.1 (cited on page 92).

Yoon Kim, Alexander Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis (June 2019). "Unsupervised Recurrent Neural Network Grammars". *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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pages 1105–1117. DOI: 10.18653/v1/N19-1114. URL: https://aclanthology.org/N19-1114 (cited on page 179).

Yoon Kim and Alexander M. Rush (Nov. 2016). "Sequence-Level Knowledge Distillation". *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pages 1317–1327. DOI: 10.18653/v1/D16-1139. URL: https://aclanthology.org/D16-1139 (cited on page 143).

Young Jin Kim and Hany Hassan (Nov. 2020). "FastFormers: Highly Efficient Transformer Models for Natural Language Understanding". *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*. Online: Association for Computational Linguistics, pages 149–158. DOI: 10.18653/v1/2020.

sustainlp-1.20. URL: https://aclanthology.org/2020.sustainlp-1.20 (cited on pages 144, 167).

Yuval Kirstain, Ori Ram, and Omer Levy (Aug. 2021). "Coreference Resolution without Span Representations". *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Online: Association for Computational Linguistics, pages 14–19. DOI: 10.18653/v1/2021.acl-short.3. URL: https://aclanthology.org/2021.acl-short.3 (cited on pages 39, 41, 44, 161, 171).

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya (2020). "Reformer: The Efficient Transformer". *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=rkgNKkHtvB (cited on page 80).

Hamidreza Kobdani and Hinrich Schütze (July 2010). "SUCRE: A Modular System for Coreference Resolution". *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden: Association for Computational Linguistics, pages 92–95. URL: https://aclanthology.org/S10-1018 (cited on page 133).

Vid Kocijan, Oana-Maria Camburu, Ana-Maria Cretu, Yordan Yordanov, Phil Blunsom, and Thomas Lukasiewicz (Nov. 2019). "WikiCREM: A Large Unsupervised Corpus for Coreference Resolution". *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Hong Kong, China: Association for Computational Linguistics, pages 4303–4312. DOI: 10.18653/v1/D19-1439. URL: https://aclanthology.org/D19-1439 (cited on pages 15, 176).

Ryuto Konno, Shun Kiyono, Yuichiroh Matsubayashi, Hiroki Ouchi, and Kentaro Inui (Nov. 2021). "Pseudo Zero Pronoun Resolution Improves Zero Anaphora Resolution". *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pages 3790–3806. DOI: 10.18653/v1/2021.emnlp-main.308. URL: https://aclanthology.org/2021.emnlp-main.308 (cited on page 11).

Satwik Kottur, Jose M. F. Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach (2018). "Visual Coreference Resolution in Visual Dialog using Neural Module Networks". *Proceedings of the European Conference on Computer Vision (ECCV)* (cited on page 6).

Jonathan K. Kummerfeld, Mohit Bansal, David Burkett, and Dan Klein (June 2011). "Mention Detection: Heuristics for the OntoNotes annotations". *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*. Portland, Oregon, USA: Association for Computational Linguistics, pages 102–106. URL: https://aclanthology.org/W11-1916 (cited on page 54).

Jonathan K. Kummerfeld and Dan Klein (Oct. 2013). "Error-Driven Analysis of Challenges in Coreference Resolution". *Proceedings of the 2013 Conference*

on Empirical Methods in Natural Language Processing. Seattle, Washington, USA: Association for Computational Linguistics, pages 265–277. URL: https://aclanthology.org/D13-1027 (cited on page 23).

Yuri Kuratov and Mikhail Arkhipov (2019). "Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language". *Computational Linguistics and Intellectual Technologies*, pages 333–339. URL: http://www.dialog-21.ru/media/4606/kuratovyplusarkhipovm-025.pdf (cited on page 115).

François Lagunas, Ella Charlaix, Victor Sanh, and Alexander M. Rush (2021). *Block Pruning For Faster Transformers*. arXiv: 2109.04838 [cs.LG] (cited on pages 149, 155).

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer (June 2016). "Neural Architectures for Named Entity Recognition". *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pages 260–270. DOI: 10.18653/v1/N16-1030. URL: https://aclanthology.org/N16-1030 (cited on page 51).

T. A. Le, M. A. Petrov, Y. M. Kuratov, and M. S. Burtsev (2019). "Sentence Level Representation and Language Models in the Task of Coreference Resolution for Russian". *Computational Linguistics and Intellectual Technologies*, pages 364–373.

URL: http://www.dialog-21.ru/media/4609/letaplusetal-160.pdf (cited on page 115).

Yann LeCun, John Denker, and Sara Solla (1990). "Optimal Brain Damage". *Advances in Neural Information Processing Systems*. Edited by D. Touretzky. Volume 2. Morgan-Kaufmann. URL: https://proceedings.neurips.cc/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf (cited on pages 143, 150).

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer (Sept. 2017). "End-to-end Neural Coreference Resolution". *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pages 188–197. DOI: 10.18653/v1/D17-1018. URL: https://aclanthology.org/D17-1018 (cited on pages 23, 27–29, 38, 41, 49, 51, 52, 56, 61, 64, 70, 73, 74, 78, 86, 92, 118).

Kenton Lee, Luheng He, and Luke Zettlemoyer (June 2018). "Higher-Order Coreference Resolution with Coarse-to-Fine Inference". *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pages 687–692. DOI: 10.18653/v1/N18-2108. URL: https://aclanthology.org/N18-2108 (cited on pages 32–34, 40, 41, 47, 52, 55, 56, 58, 70, 97, 115, 118, 133).

Hector J. Levesque, Ernest Davis, and L. Morgenstern (2011). "The Winograd Schema Challenge". *KR* (cited on page 15).

Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston (2017). "Learning through Dialogue Interactions by Asking Questions". *ICLR* (cited on page 91).

Manling Li, Ying Lin, Joseph Hoover, Spencer Whitehead, Clare Voss, Morteza Dehghani, and Heng Ji (June 2019). "Multilingual Entity, Relation, Event and Human Value Extraction". *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Minneapolis, Minnesota: Association for Computational Linguistics, pages 110–115. DOI: 10.18653/v1/N19-4019. URL: https://aclanthology.org/N19-4019 (cited on page 111).

Manling Li, Alireza Zareian, Ying Lin, Xiaoman Pan, Spencer Whitehead, Brian Chen, Bo Wu, Heng Ji, Shih-Fu Chang, Clare Voss, Daniel Napierski, and Marjorie Freedman (July 2020a). "GAIA: A Fine-grained Multimedia Knowledge Extraction System". *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Online: Association for Computational Linguistics, pages 77–86. DOI: 10.18653/v1/2020.acl-demos.11. URL: https://aclanthology.org/2020.acl-demos.11 (cited on pages 110, 112).

Pengshuai Li, Xinsong Zhang, Weijia Jia, and Wei Zhao (Nov. 2020b). "Active Testing: An Unbiased Evaluation Method for Distantly Supervised Relation Extraction". *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pages 204–211. DOI: 10.18653/v1/2020.

findings-emnlp.20. URL: https://aclanthology.org/2020.findings-emnlp.20 (cited on page 176).

Xiang Lisa Li and Percy Liang (Aug. 2021). "Prefix-Tuning: Optimizing Continuous Prompts for Generation". *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pages 4582–4597. DOI: 10.18653/v1/2021.acl-long.353. URL: https://aclanthology.org/2021.acl-long.353 (cited on page 160).

Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joey Gonzalez (2020c). "Train Big, Then Compress: Rethinking Model Size for Efficient Training and Inference of Transformers". *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Volume 119. Proceedings of Machine Learning Research. PMLR, pages 5958–5968. URL: http://proceedings.mlr.press/v119/li20m.html (cited on page 167).

Vladislav Lialin, Rahul Goel, Andrey Simanovsky, Anna Rumshisky, and Rushin Shah (2021). *Update Frequently, Update Fast: Retraining Semantic Parsing Systems in a Fraction of Time*. arXiv: 2010.07865 [cs.CL] (cited on page 122).

Chen Liang, Simiao Zuo, Minshuo Chen, Haoming Jiang, Xiaodong Liu, Pengcheng He, Tuo Zhao, and Weizhu Chen (Aug. 2021). "Super Tickets in Pre-Trained Language Models: From Model Compression to Improving Generalization". *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th*

International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Online: Association for Computational Linguistics, pages 6524–6538. DOI: 10.18653/v1/2021.acl-long.510. URL: https://aclanthology.org/2021.acl-long.510 (cited on page 151).

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu (July 2020). "A Joint Neural Model for Information Extraction with Global Features". *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pages 7999–8009. DOI: 10.18653/v1/2020.acl-main.713. URL: https://aclanthology.org/2020.acl-main.713 (cited on page 63).

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao (July 2019). "Multi-Task Deep Neural Networks for Natural Language Understanding". *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pages 4487–4496. DOI: 10.18653/v1/P19-1441. URL: https://aclanthology.org/P19-1441 (cited on pages 147, 160).

Pengcheng Lu and Massimo Poesio (Nov. 2021). "Coreference Resolution for the Biomedical Domain: A Survey". *Proceedings of the Fourth Workshop on Computational Models of Reference, Anaphora and Coreference*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pages 12–23. DOI: 10.18653/v1/2021.crac-1.2. URL: https://aclanthology.org/2021.crac-1.2 (cited on page 9).

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi (2018). "Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction". *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pages 3219–3232. DOI: 10.18653/v1/D18-1360. URL: https://aclanthology.org/D18-1360 (cited on page 51).

Xiaoqiang Luo (Oct. 2005). "On Coreference Resolution Performance Metrics". *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pages 25–32. URL: https://aclanthology.org/H05-1004 (cited on pages 20, 21, 77).

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos (July 2004). "A Mention-Synchronous Coreference Resolution Algorithm Based On the Bell Tree". *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*. Barcelona, Spain, pages 135–142. DOI: 10.3115/1218955.1218973. URL: https://aclanthology.org/P04-1018 (cited on page 24).

Minh-Thang Luong and Christopher D. Manning (2015). "Stanford Neural Machine Translation Systems for Spoken Language Domain". *International Workshop on Spoken Language Translation* (cited on pages 121, 122).

Xutai Ma, Juan Pino, and Philipp Koehn (Dec. 2020). "SimulMT to SimulST: Adapting Simultaneous Text Translation to End-to-End Simultaneous Speech Translation". *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, pages 582–587. URL: https://aclanthology.org/2020.aacl-main.58 (cited on page 91).

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky (June 2014). "The Stanford CoreNLP Natural Language Processing Toolkit". *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, Maryland: Association for Computational Linguistics, pages 55–60. DOI: 10.3115/v1/P14-5010. URL: https://aclanthology.org/P14-5010 (cited on page 120).

Michael Mathioudakis and Nick Koudas (2010). "TwitterMonitor: Trend Detection over the Twitter Stream". *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*. SIGMOD '10. Indianapolis, Indiana, USA: Association for Computing Machinery, 1155–1158. ISBN: 9781450300322. DOI: 10.1145/1807167.1807306. URL: https://doi.org/10.1145/1807167.1807306 (cited on page 91).

Michael McCloskey and Neal J. Cohen (1989). "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem". Edited by Gordon

H. Bower. Volume 24. Psychology of Learning and Motivation. Academic Press, pages 109–165. DOI: https://doi.org/10.1016/S0079-7421(08)60536-8. URL: https://www.sciencedirect.com/science/article/pii/S0079742108605368 (cited on page 136).

Paul Michel, Omer Levy, and Graham Neubig (2019). "Are Sixteen Heads Really Better than One?" *Advances in Neural Information Processing Systems*. Edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Volume 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2019/file/2c601ad9d2ff9bc8b282670cdd54f69f-Paper.pdf (cited on page 143).

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory F. Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu (2018). "Mixed Precision Training". *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=r1gs9JgRZ (cited on page 144).

Nafise Sadat Moosavi and Michael Strube (Aug. 2016). "Which Coreference Evaluation Metric Do You Trust? A Proposal for a Link-based Entity Aware Metric". *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pages 632–642. DOI: 10.18653/v1/P16-1060. URL: https://aclanthology.org/P16-1060 (cited on pages 19, 20, 22).

Nafise Sadat Moosavi and Michael Strube (2018). "Using Linguistic Features to Improve the Generalization Capability of Neural Coreference Resolvers". *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pages 193–203. DOI: 10.18653/v1/D18-1018. URL: https://aclanthology.org/D18-1018 (cited on page 121).

Christoph Müller and Michael Strube (2006). "Multi-level annotation of linguistic data with MMAX2". *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*. Edited by Sabine Braun, Kurt Kohn, and Joybrato Mukherjee. Frankfurt a.M., Germany: Peter Lang, pages 197–214 (cited on page 16).

Kenton Murray and David Chiang (Sept. 2015). "Auto-Sizing Neural Networks: With Applications to n-gram Language Models". *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pages 908–916. DOI: 10.18653/v1/D15-1107. URL: https://aclanthology.org/D15-1107 (cited on page 143).

Kenton Murray, Brian DuSell, and David Chiang (Nov. 2019). "Efficiency through Auto-Sizing: Notre Dame NLP's Submission to the WNGT 2019 Efficiency Task". *Proceedings of the 3rd Workshop on Neural Generation and Translation*. Hong Kong: Association for Computational Linguistics, pages 297–301. DOI: 10.18653/v1/D19-5634. URL: https://aclanthology.org/D19-5634 (cited on page 143).

Judith Muzerelle, Anaïs Lefeuvre, Emmanuel Schang, Jean-Yves Antoine, Aurore Pelletier, Denis Maurel, Iris Eshkol, and Jeanne Villaneau (May 2014).

"ANCOR_Centre, a large free spoken French coreference corpus: description of the resource and reliability measures". *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), pages 843–847. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/150_Paper.pdf (cited on page 14).

Vincent Ng (July 2010). "Supervised Noun Phrase Coreference Research: The First Fifteen Years". *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, pages 1396–1411. URL: https://aclanthology.org/P10-1142 (cited on page 24).

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman (May 2016). "Universal Dependencies v1: A Multilingual Treebank Collection". *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), pages 1659–1666. URL: https://aclanthology.org/L16-1262 (cited on page 14).

Bruno Oberle (May 2018). "SACR: A Drag-and-Drop Based Tool for Coreference Annotation". *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language

Resources Association (ELRA). URL: https://aclanthology.org/L18-1059 (cited on page 16).

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura (June 2014). "Optimizing Segmentation Strategies for Simultaneous Speech Translation". *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pages 551–556. DOI: 10.3115/v1/P14-2090. URL: https://aclanthology.org/P14-2090 (cited on page 91).

Shon Otmazgin, Arie Cattan, and Yoav Goldberg (2022). *LingMess: Linguistically Informed Multi Expert Scorers for Coreference Resolution*. DOI: 10.48550/ARXIV.2205.12644. URL: https://arxiv.org/abs/2205.12644 (cited on page 24).

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji (July 2017). "Cross-lingual Name Tagging and Linking for 282 Languages". *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pages 1946–1958. DOI: 10.18653/v1/P17-1178. URL: https://aclanthology.org/P17-1178 (cited on page 112).

Hao Peng, Sam Thomson, Swabha Swayamdipta, and Noah A. Smith (June 2018). "Learning Joint Semantic Parsers from Disjoint Data". *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans,

Louisiana: Association for Computational Linguistics, pages 1492–1502. DOI: 10.18653/v1/N18-1135. URL: https://aclanthology.org/N18-1135 (cited on page 51).

Nanyun Peng, Francis Ferraro, Mo Yu, Nicholas Andrews, Jay DeYoung, Max Thomas, Matthew R. Gormley, Travis Wolfe, Craig Harman, Benjamin Van Durme, and Mark Dredze (June 2015). "A Concrete Chinese NLP Pipeline". *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Denver, Colorado: Association for Computational Linguistics, pages 86–90. DOI: 10.3115/v1/N15-3018. URL: https://aclanthology.org/N15-3018 (cited on page 112).

Jeffrey Pennington, Richard Socher, and Christopher Manning (Oct. 2014). "GloVe: Global Vectors for Word Representation". *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pages 1532–1543. DOI: 10.3115/v1/D14-1162. URL: https://aclanthology.org/D14-1162 (cited on page 30).

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (June 2018). "Deep Contextualized Word Representations". *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pages 2227–2237. DOI: 10.18653/v1/N18-1202. URL: https://aclanthology.org/N18-1202 (cited on pages 30, 47).

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein (July 2006). "Learning Accurate, Compact, and Interpretable Tree Annotation". *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia: Association for Computational Linguistics, pages 433–440. DOI: 10.3115/1220175.1220230. URL: https://aclanthology.org/P06-1055 (cited on page 55).

Massimo Poesio and Ron Artstein (May 2008). "Anaphoric Annotation in the ARRAU Corpus". *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Marrakech, Morocco: European Language Resources Association (ELRA). URL: http://www.lrec-conf.org/proceedings/lrec2008/pdf/297_paper.pdf (cited on pages 12, 53, 124).

Massimo Poesio, Yulia Grishina, Varada Kolhatkar, Nafise Moosavi, Ina Roesiger, Adam Roussel, Fabian Simonjetz, Alexandra Uma, Olga Uryupina, Juntao Yu, and Heike Zinsmeister (June 2018). "Anaphora Resolution with the ARRAU Corpus". *Proceedings of the First Workshop on Computational Models of Reference, Anaphora and Coreference*. New Orleans, Louisiana: Association for Computational Linguistics, pages 11–22. DOI: 10.18653/v1/W18-0702. URL: https://aclanthology.org/W18-0702 (cited on pages 9, 10, 125).

Massimo Poesio and Hannes Rieser (2011). "An incremental model of anaphora and reference resolution based on resource situations." *Dialogue Discourse* 2, pages 235–277 (cited on page 94).

Corbèn Poot and Andreas van Cranenburgh (Dec. 2020). "A Benchmark of Rule-Based and Neural Coreference Resolution in Dutch Novels and News". *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*. Barcelona, Spain (online): Association for Computational Linguistics, pages 79–90. URL: https://aclanthology.org/2020.crac-1.9 (cited on page 119).

Sameer Pradhan, Eduard H. Hovy, Mitchell P. Marcus, Martha Palmer, Lance A. Ramshaw, and Ralph M. Weischedel (2007). "OntoNotes: A Unified Relational Semantic Representation". *International Conference on Semantic Computing (ICSC 2007)*, pages 517–526 (cited on pages 9, 14, 53).

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong (Aug. 2013). "Towards Robust Linguistic Analysis using OntoNotes". *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. Sofia, Bulgaria: Association for Computational Linguistics, pages 143–152. URL: https://aclanthology.org/W13-3516 (cited on pages 50, 76, 95, 114).

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang (July 2012). "CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes". *Joint Conference on EMNLP and CoNLL - Shared Task*. Jeju Island, Korea: Association for Computational Linguistics, pages 1–40. URL: https://aclanthology.org/W12-4501 (cited on pages 11, 12, 14, 17, 19, 22, 41).

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2020). "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". *ArXiv* abs/1910.10683 (cited on page 43).

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning (Oct. 2010). "A Multi-Pass Sieve for Coreference Resolution". *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Cambridge, MA: Association for Computational Linguistics, pages 492–501. URL: https://aclanthology.org/D10-1048 (cited on pages 24, 26, 38, 48, 50, 70, 133).

Altaf Rahman and Vincent Ng (Aug. 2009). "Supervised Models for Coreference Resolution". *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, pages 968–977. URL: https://aclanthology.org/D09-1101 (cited on page 24).

Altaf Rahman and Vincent Ng (July 2012). "Resolving Complex Cases of Definite Pronouns: The Winograd Schema Challenge". *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island, Korea: Association for Computational Linguistics, pages 777–789. URL: https://aclanthology.org/D12-1071 (cited on page 15).

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang (Nov. 2016). "SQuAD: 100,000+ Questions for Machine Comprehension of Text". *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pages 2383–2392. DOI: 10.18653/v1/D16-1264. URL: https://aclanthology.org/D16-1264 (cited on pages 152, 158).

James Ravenscroft, Amanda Clare, Arie Cattan, Ido Dagan, and Maria Liakata (Apr. 2021). "CDˆ2CR: Co-reference resolution across documents and domains". *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, pages 270–280. DOI: 10.18653/v1/2021.eacl-main.21. URL: https://aclanthology.org/2021.eacl-main.21 (cited on page 16).

M. Recasens and E. Hovy (2011). "BLANC: Implementing the Rand index for coreference evaluation". *Natural Language Engineering* 17.4, 485–510. DOI: 10.1017/S135132491000029X (cited on page 22).

Marta Recasens, Lluís Màrquez, Emili Sapena, M. Antònia Martí, Mariona Taulé, Véronique Hoste, Massimo Poesio, and Yannick Versley (July 2010). "SemEval-2010 Task 1: Coreference Resolution in Multiple Languages". *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden: Association for Computational Linguistics, pages 1–8. URL: https://aclanthology.org/S10-1001 (cited on pages 9, 14, 114, 125, 177).

Marta Recasens and Maria Antònia Martí (2010). "AnCora-CO: Coreferentially annotated corpora for Spanish and Catalan". *Language Resources and Evaluation* 44, pages 315–345 (cited on page 14).

Kepa Joseba Rodríguez, Francesca Delogu, Yannick Versley, Egon W. Stemle, and Massimo Poesio (May 2010). "Anaphoric Annotation of Wikipedia and Blogs in the Live Memories Corpus". *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA). URL: http://www.lrec-conf.org/proceedings/lrec2010/pdf/431_Paper.pdf (cited on page 14).

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme (June 2018). "Gender Bias in Coreference Resolution". *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pages 8–14. DOI: 10.18653/v1/N18-2002. URL: https://aclanthology.org/N18-2002 (cited on pages 15, 119).

Tara N. Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran (2013). "Low-rank matrix factorization for Deep Neural Network training with high-dimensional output targets". *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6655–6659 (cited on page 144).

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf (2020a). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. arXiv: 1910.01108 [cs.CL] (cited on pages 143, 146, 161).

Victor Sanh, Thomas Wolf, and Alexander M. Rush (2020b). "Movement Pruning: Adaptive Sparsity by Fine-Tuning". *NeurIPS*. URL: https://proceedings.neurips.cc/paper/2020/hash/eae15aabaa768ae4a5993a8a4f4fa6e4-Abstract.html (cited on pages 143, 144, 150–152, 154, 163, 165).

Timo Schick and Hinrich Schütze (June 2021). "It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners". *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pages 2339–2352. DOI: 10.18653/v1/2021.naacl-main.185. URL: https://aclanthology.org/2021.naacl-main.185 (cited on pages 119, 130).

David Schlangen, Timo Baumann, and Michaela Atterer (Sept. 2009). "Incremental Reference Resolution: The Task, Metrics for Evaluation, and a Bayesian Filtering Model that is Sensitive to Disfluencies". *Proceedings of the SIGDIAL 2009 Conference*. London, UK: Association for Computational Linguistics, pages 30–37. URL: https://aclanthology.org/W09-3905 (cited on pages 92, 94, 107).

Fynn Schröder, Hans Ole Hatzel, and Chris Biemann (2021). "Neural End-to-end Coreference Resolution for German in Different Domains". *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021)*. Düsseldorf,

Germany: KONVENS 2021 Organizers, pages 170–181. URL: https://aclanthology.org/2021.konvens-1.15 (cited on page 171).

Abigail See, Minh-Thang Luong, and Christopher D. Manning (Aug. 2016). "Compression of Neural Machine Translation Models via Pruning". *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, pages 291–301. DOI: 10.18653/v1/K16-1029. URL: https://aclanthology.org/K16-1029 (cited on page 143).

Rico Sennrich, Barry Haddow, and Alexandra Birch (Aug. 2016). "Improving Neural Machine Translation Models with Monolingual Data". *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pages 86–96. DOI: 10.18653/v1/P16-1009. URL: https://aclanthology.org/P16-1009 (cited on page 119).

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer (2020). "Q-BERT: Hessian Based Ultra Low Precision Quantization of BERT". *Proceedings of the AAAI Conference on Artificial Intelligence* 34.05, pages 8815–8821. DOI: 10.1609/aaai.v34i05.6409. URL: https://ojs.aaai.org/index.php/AAAI/article/view/6409 (cited on page 144).

Tomohide Shibata and Sadao Kurohashi (July 2018). "Entity-Centric Joint Modeling of Japanese Coreference Resolution and Predicate Argument Structure Analysis".

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics, pages 579–589. DOI: 10.18653/v1/P18-1054. URL: https://aclanthology.org/P18-1054 (cited on page 11).

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh (Nov. 2020). "AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pages 4222–4235. DOI: 10.18653/v1/2020.emnlp-main.346. URL: https://aclanthology.org/2020.emnlp-main.346 (cited on page 146).

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts (Oct. 2013). "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank". *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pages 1631–1642. URL: https://aclanthology.org/D13-1170 (cited on page 158).

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii (Apr. 2012). "brat: a Web-based Tool for NLP-Assisted Text Annotation". *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Avignon,

France: Association for Computational Linguistics, pages 102–107. URL: https://aclanthology.org/E12-2021 (cited on page 16).

Scott C. Stoness, Joel Tetreault, and James Allen (July 2004). "Incremental Parsing with Reference Interaction". *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*. Barcelona, Spain: Association for Computational Linguistics, pages 18–25. URL: https://aclanthology.org/W04-0304 (cited on pages 92, 94).

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang (2021). "ERNIE 3.0: Large-scale Knowledge Enhanced Pre-training for Language Understanding and Generation". *ArXiv* abs/2107.02137 (cited on page 160).

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang (2020a). "ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding". *Proceedings of the AAAI Conference on Artificial Intelligence* 34.05, pages 8968–8975. DOI: 10.1609/aaai.v34i05.6428. URL: https://ojs.aaai.org/index.php/AAAI/article/view/6428 (cited on page 160).

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou (July 2020b). "MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices". *Proceedings of the 58th Annual Meeting of the Association for*

Computational Linguistics. Online: Association for Computational Linguistics, pages 2158–2170. DOI: 10.18653/v1/2020.acl-main.195. URL: https://aclanthology.org/2020.acl-main.195 (cited on pages 146, 152).

Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith (2018). "Syntactic Scaffolds for Semantic Structures". *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pages 3772–3782. DOI: 10.18653/v1/D18-1412. URL: https://aclanthology.org/D18-1412 (cited on pages 36, 42, 51).

Ian Tenney, Dipanjan Das, and Ellie Pavlick (July 2019a). "BERT Rediscovers the Classical NLP Pipeline". *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pages 4593–4601. DOI: 10.18653/v1/P19-1452. URL: https://aclanthology.org/P19-1452 (cited on pages 15, 43, 138).

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick (2019b). "What do you learn from context? Probing for sentence structure in contextualized word representations". *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=SJzSgnRcKX (cited on pages 15, 43, 51).

Raghuveer Thirukovalluru, Nicholas Monath, Kumar Shridhar, Manzil Zaheer, Mrinmaya Sachan, and Andrew McCallum (Aug. 2021). "Scaling Within Document Coreference to Long Texts". *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, pages 3921–3931. DOI: 10.18653/v1/2021.findings-acl.343. URL: https://aclanthology.org/2021.findings-acl.343 (cited on pages 132, 133).

Hariprasad Timmapathini, Anmol Nayak, Sarathchandra Mandadi, Siva Sangada, Vaibhav Kesri, Karthikeyan Ponnalagu, and Vijendran Gopalan Venkoparao (2021). "Probing the SpanBERT Architecture to interpret Scientific Domain Adaptation Challenges for Coreference Resolution". *Proceedings of the Workshop on Scientific Document Understanding co-located with 35th AAAI Conference on Artificial Inteligence, SDU@AAAI 2021, Virtual Event, February 9, 2021*. Edited by Amir Pouran Ben Veyseh, Franck Dernoncourt, Thien Huu Nguyen, Walter Chang, and Leo Anthony Celi. Volume 2831. CEUR Workshop Proceedings. CEUR-WS.org. URL: http://ceur-ws.org/Vol-2831/paper10.pdf (cited on page 119).

S Toldova, A. Roytberg, Alina Ladygina, Maria Vasilyeva, Ilya Azerkovich, Matvei Kurzukov, G. Sim, D.V. Gorshkov, A. Ivanova, Anna Nedoluzhko, and Y. Grishina (Jan. 2014). "RU-EVAL-2014: Evaluating anaphora and coreference resolution for Russian". *Computational Linguistics and Intellectual Technologies*, pages 681–694. URL: http://www.dialog-21.ru/digests/dialog2014/materials/pdf/ToldovaSJu.pdf (cited on pages 14, 114).

Shubham Toshniwal, Haoyue Shi, Bowen Shi, Lingyu Gao, Karen Livescu, and Kevin Gimpel (July 2020a). "A Cross-Task Analysis of Text Span Representations". *Proceedings of the 5th Workshop on Representation Learning for NLP*. Online: Association for Computational Linguistics, pages 166–176. DOI: 10.18653/v1/2020.repl4nlp-1.20. URL: https://aclanthology.org/2020.repl4nlp-1.20 (cited on pages 37, 59).

Shubham Toshniwal, Sam Wiseman, Allyson Ettinger, Karen Livescu, and Kevin Gimpel (Nov. 2020b). "Learning to Ignore: Long Document Coreference with Bounded Memory Neural Networks". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pages 8519–8526. DOI: 10.18653/v1/2020.emnlp-main.685. URL: https://aclanthology.org/2020.emnlp-main.685 (cited on pages 35, 38, 89).

Shubham Toshniwal, Patrick Xia, Sam Wiseman, Karen Livescu, and Kevin Gimpel (Nov. 2021). "On Generalization in Coreference Resolution". *Proceedings of the Fourth Workshop on Computational Models of Reference, Anaphora and Coreference*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pages 111–120. DOI: 10.18653/v1/2021.crac-1.12. URL: https://aclanthology.org/2021.crac-1.12 (cited on pages 37, 48, 59, 96, 142, 160, 171).

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio (July 2010). "Word Representations: A Simple and General Method for Semi-Supervised Learning".

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Uppsala, Sweden: Association for Computational Linguistics, pages 384–394. URL: https://aclanthology.org/P10-1040 (cited on page 30).

Gorka Urbizu, Ander Soraluze, and Olatz Arregi (Dec. 2020). "Sequence to Sequence Coreference Resolution". *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*. Barcelona, Spain (online): Association for Computational Linguistics, pages 39–46. URL: https://aclanthology.org/2020.crac-1.5 (cited on page 129).

Olga Uryupina, Ron Artstein, Antonella Bristot, Federica Cavicchio, Francesca Delogu, Kepa J. Rodriguez, and Massimo Poesio (2020). "Annotating a broad range of anaphoric phenomena, in a variety of genres: the ARRAU Corpus". *Natural Language Engineering* 26.1, 95–128. DOI: 10.1017/S1351324919000056 (cited on pages 12, 124).

Olga Uryupina, Ron Artstein, Antonella Bristot, Federica Cavicchio, Kepa Rodriguez, and Massimo Poesio (May 2016). "ARRAU: Linguistically-Motivated Annotation of Anaphoric Descriptions". *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), pages 2058–2062. URL: https://aclanthology.org/L16-1326 (cited on page 12).

Andreas van Cranenburgh (2019). "A Dutch coreference resolution system with an evaluation on literary fiction". *Computational Linguistics in the Netherlands*

Journal 9, pages 27–54. URL: https://clinjournal.org/clinj/article/view/91 (cited on pages 11, 14).

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman (1995). "A Model-Theoretic Coreference Scoring Scheme". *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*. URL: https://aclanthology.org/M95-1005 (cited on pages 19, 77).

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman (2019a). "SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems". *NeurIPS* (cited on page 15).

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman (2019b). "SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems". *arXiv preprint 1905.00537* (cited on page 43).

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman (2019). "Neural Network Acceptability Judgments". *Transactions of the Association for Computational Linguistics* 7, pages 625–641. DOI: 10.1162/tacl_a_00290. URL: https://aclanthology.org/Q19-1040 (cited on page 158).

Bonnie Lynn Webber (June 1988). "Discourse Deixis: Reference to Discourse Segments". *26th Annual Meeting of the Association for Computational Linguistics*. Buffalo,

New York, USA: Association for Computational Linguistics, pages 113–122. DOI: 10.3115/982023.982037. URL: https://aclanthology.org/P88-1014 (cited on page 175).

Kellie Webster and James R. Curran (Aug. 2014). "Limited memory incremental coreference resolution". *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, pages 2129–2139. URL: https://aclanthology.org/C14-1201 (cited on pages 37, 71).

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, pages 605–617. DOI: 10.1162/tacl_a_00240. URL: https://aclanthology.org/Q18-1042 (cited on pages 15, 119).

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus (2022). "Emergent Abilities of Large Language Models". *ArXiv* abs/2206.07682 (cited on page 167).

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. (2013). "OntoNotes Release 5.0". *Linguistic Data Consortium, Philadelphia, PA*. DOI: 10.35111/xmhb-2b84 (cited on pages 11, 14, 50, 76, 114, 120, 123, 160).

Adina Williams, Nikita Nangia, and Samuel Bowman (June 2018). "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference". *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pages 1112–1122. DOI: 10.18653/v1/N18-1101. URL: https://aclanthology.org/N18-1101 (cited on page 152).

Sam Wiseman, Alexander M. Rush, Stuart Shieber, and Jason Weston (July 2015). "Learning Anaphoricity and Antecedent Ranking Features for Coreference Resolution". *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, pages 1416–1426. DOI: 10.3115/v1/P15-1137. URL: https://aclanthology.org/P15-1137 (cited on page 25).

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber (June 2016). "Learning Global Features for Coreference Resolution". *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pages 994–1004. DOI: 10.18653/v1/N16-1114. URL: https://aclanthology.org/N16-1114 (cited on pages 25, 41, 49, 70).

Shijie Wu and Mark Dredze (Nov. 2019). "Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT". *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pages 833–844. DOI: 10.18653/v1/D19-1077. URL: https://aclanthology.org/D19-1077 (cited on page 115).

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li (July 2020). "CorefQA: Coreference Resolution as Query-based Span Prediction". *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pages 6953–6963. DOI: 10.18653/v1/2020.acl-main.622. URL: https://aclanthology.org/2020.acl-main.622 (cited on pages 3, 35, 39–41, 70, 74, 77, 78, 92, 96, 118).

Zhaofeng Wu and Matt Gardner (Nov. 2021). "Understanding Mention Detector-Linker Interaction in Neural Coreference Resolution". *Proceedings of the Fourth Workshop on Computational Models of Reference, Anaphora and Coreference*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pages 150–157. DOI: 10.18653/v1/2021.crac-1.16. URL: https://aclanthology.org/2021.crac-1.16 (cited on pages 43, 133).

Patrick Xia, Guanghui Qin, Siddharth Vashishtha, Yunmo Chen, Tongfei Chen, Chandler May, Craig Harman, Kyle Rawlins, Aaron Steven White, and Benjamin Van Durme (Apr. 2021). "LOME: Large Ontology Multilingual Extraction".

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations. Online: Association for Computational Linguistics, pages 149–159. DOI: 10.18653/v1/2021.eacl-demos.19. URL: https://aclanthology.org/2021.eacl-demos.19 (cited on pages 60, 64, 110, 111, 115, 116).

Patrick Xia, João Sedoc, and Benjamin Van Durme (Nov. 2020a). "Incremental Neural Coreference Resolution in Constant Memory". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pages 8617–8624. DOI: 10.18653/v1/2020.emnlp-main.695. URL: https://aclanthology.org/2020.emnlp-main.695 (cited on pages 38, 93, 97, 122, 128).

Patrick Xia, Elias Stengel-Eskin, Tongfei Chen, Seth Ebner, Nils Holzenberger, Ryan Culkin, Pushpendre Rastogi, Xutai Ma, and Benjamin Van Durme (2018). "NIST TAC SM-KBP 2018 System Description: JHU/UR Pipeline". *Theory and Applications of Categories* (cited on page 61).

Patrick Xia and Benjamin Van Durme (Nov. 2021). "Moving on from OntoNotes: Coreference Resolution Model Transfer". *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pages 5241–5256. DOI: 10.18653/v1/2021.emnlp-main.425. URL: https://aclanthology.org/2021.emnlp-main.425 (cited on pages 13, 37, 59, 96, 105).

Patrick Xia and Benjamin Van Durme (Oct. 2022). "Online Neural Coreference Resolution with Rollback". *Proceedings of the Fifth Workshop on Computational Models of Reference, Anaphora and Coreference*. Gyeongju, Republic of Korea: Association for Computational Linguistics, pages 13–21. URL: https://aclanthology.org/2022.crac-1.2 (cited on page 38).

Patrick Xia, Shijie Wu, and Benjamin Van Durme (Nov. 2020b). "Which *BERT? A Survey Organizing Contextualized Encoders". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pages 7516–7533. DOI: 10.18653/v1/2020.emnlp-main.608. URL: https://aclanthology.org/2020.emnlp-main.608 (cited on page 34).

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin (July 2020). "DeeBERT: Dynamic Early Exiting for Accelerating BERT Inference". *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pages 2246–2251. DOI: 10.18653/v1/2020.acl-main.204. URL: https://aclanthology.org/2020.acl-main.204 (cited on page 167).

Canwen Xu and Julian McAuley (2022). "A Survey on Model Compression for Natural Language Processing". *ArXiv* abs/2202.07105 (cited on page 143).

Haoran Xu, Seth Ebner, Mahsa Yarmohammadi, Aaron Steven White, Benjamin Van Durme, and Kenton Murray (Apr. 2021). "Gradual Fine-Tuning for Low-Resource

Domain Adaptation". *Proceedings of the Second Workshop on Domain Adaptation for NLP*. Kyiv, Ukraine: Association for Computational Linguistics, pages 214–221. URL: https://aclanthology.org/2021.adaptnlp-1.22 (cited on page 122).

Liyan Xu and Jinho D. Choi (Nov. 2020). "Revealing the Myth of Higher-Order Inference in Coreference Resolution". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pages 8527–8533. DOI: 10.18653/v1/2020.emnlp-main.686. URL: https://aclanthology.org/2020.emnlp-main.686 (cited on pages 33, 34, 37, 40, 41, 59, 70, 93, 97).

Liyan Xu and Jinho D. Choi (Nov. 2021). "Adapted End-to-End Coreference Resolution System for Anaphoric Identities in Dialogues". *Proceedings of the CODI-CRAC 2021 Shared Task on Anaphora, Bridging, and Discourse Deixis in Dialogue*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pages 55–62. DOI: 10.18653/v1/2021.codi-sharedtask.6. URL: https://aclanthology.org/2021.codi-sharedtask.6 (cited on page 97).

Liyan Xu and Jinho D. Choi (July 2022). "Online Coreference Resolution for Dialogue Processing: Improving Mention-Linking on Real-Time Conversations". *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*. Seattle, Washington: Association for Computational Linguistics, pages 341–347. DOI: 10.18653/v1/2022.starsem-1.30. URL: https://aclanthology.org/2022.starsem-1.30 (cited on pages 38, 92).

Mingbin Xu, Hui Jiang, and Sedtawut Watcharawittayakul (July 2017). "A Local Detection Approach for Named Entity Recognition and Mention Detection". *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pages 1237–1247. DOI: 10.18653/v1/P17-1114. URL: https://aclanthology.org/P17-1114 (cited on page 51).

Huanrui Yang, Minxue Tang, Wei Wen, Feng Yan, Daniel Hu, Ang Li, Hai Li, and Yiran Chen (2020). "Learning Low-Rank Deep Neural Networks via Singular Vector Orthogonality Regularization and Singular Value Sparsification". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (cited on pages 149, 150).

Jian Bo Yang, Qi Mao, Qiao Liang Xiang, Ivor Wai-Hung Tsang, Kian Ming Adam Chai, and Hai Leong Chieu (July 2012). "Domain Adaptation for Coreference Resolution: An Adaptive Ensemble Approach". *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island, Korea: Association for Computational Linguistics, pages 744–753. URL: https://aclanthology.org/D12-1068 (cited on page 121).

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu (Nov. 2020). "Coreferential Reasoning Learning for Language Representation". *Proceedings of the 2020 Conference on Empirical Methods in*

Natural Language Processing (EMNLP). Online: Association for Computational Linguistics, pages 7170–7186. DOI: 10.18653/v1/2020.emnlp-main.582. URL: https://aclanthology.org/2020.emnlp-main.582 (cited on page 176).

Juntao Yu, Nafise Sadat Moosavi, Silviu Paun, and Massimo Poesio (Dec. 2020a). "Free the Plural: Unrestricted Split-Antecedent Anaphora Resolution". *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, pages 6113–6125. DOI: 10.18653/v1/2020.coling-main.538. URL: https://aclanthology.org/2020.coling-main.538 (cited on page 125).

Juntao Yu, Nafise Sadat Moosavi, Silviu Paun, and Massimo Poesio (June 2021). "Stay Together: A System for Single and Split-antecedent Anaphora Resolution". *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pages 4174–4184. DOI: 10.18653/v1/2021.naacl-main.329. URL: https://aclanthology.org/2021.naacl-main.329 (cited on pages 9, 125).

Juntao Yu, Alexandra Uma, and Massimo Poesio (May 2020b). "A Cluster Ranking Model for Full Anaphora Resolution". English. *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pages 11–20. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.2 (cited on pages 38, 133).

Jiahong Yuan, Mark Liberman, and Christopher Cieri (2006). "Towards an integrated understanding of speaking rate in conversation". *Ninth International Conference on Spoken Language Processing* (cited on page 104).

Michelle Yuan, Patrick Xia, Chandler May, Benjamin Van Durme, and Jordan Boyd-Graber (2021). *Adapting Coreference Resolution Models through Active Learning*. DOI: 10.48550/ARXIV.2104.07611. URL: https://arxiv.org/abs/2104.07611 (cited on pages 17, 59, 171, 176).

Zdeněk Žabokrtský, Miloslav Konopík, Anna Nedoluzhko, Michal Novák, Maciej Ogrodniczuk, Martin Popel, Ondřej Pražák, Jakub Sido, Daniel Zeman, and Yilun Zhu (Oct. 2022). "Findings of the Shared Task on Multilingual Coreference Resolution". *Proceedings of the CRAC 2022 Shared Task on Multilingual Coreference Resolution*. Gyeongju, Republic of Korea: Association for Computational Linguistics, pages 1–17. URL: https://aclanthology.org/2022.crac-mcr.1 (cited on pages 11, 14).

Rui Zhang, Cícero Nogueira dos Santos, Michihiro Yasunaga, Bing Xiang, and Dragomir Radev (July 2018a). "Neural Coreference Resolution with Deep Biaffine Attention by Joint Mention Detection and Mention Clustering". *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, pages 102–107. DOI: 10.18653/v1/P18-2017. URL: https://aclanthology.org/P18-2017 (cited on pages 36, 52, 55).

Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou (2018b). "Neural Latent Extractive Document Summarization". *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pages 779–784. DOI: 10.18653/v1/D18-1088. URL: https://aclanthology.org/D18-1088 (cited on page 88).

Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yao, Sanjeev Khudanpur, and James Glass (2016). "Highway long short-term memory rnns for distant speech recognition". *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pages 5755–5759 (cited on page 94).

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang (June 2019). "Gender Bias in Contextualized Word Embeddings". *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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pages 629–634. DOI: 10.18653/v1/N19-1064. URL: https://aclanthology.org/N19-1064 (cited on page 119).

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang (June 2018). "Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods". *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational

Linguistics, pages 15–20. DOI: 10.18653/v1/N18-2003. URL: https://aclanthology.org/N18-2003 (cited on page 15).

Shanheng Zhao and Hwee Tou Ng (Apr. 2014). "Domain Adaptation with Active Learning for Coreference Resolution". *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi)*. Gothenburg, Sweden: Association for Computational Linguistics, pages 21–29. DOI: 10.3115/v1/W14-1104. URL: https://aclanthology.org/W14-1104 (cited on page 121).

Boyuan Zheng, Patrick Xia, Mahsa Yarmohammadi, and Benjamin Van Durme (2022). *Multilingual Coreference Resolution in Multiparty Dialogue*. DOI: 10.48550/ARXIV.2208.01307. URL: https://arxiv.org/abs/2208.01307 (cited on page 175).

Ethan Zhou and Jinho D. Choi (Aug. 2018). "They Exist! Introducing Plural Mentions to Coreference Resolution and Entity Linking". *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pages 24–34. URL: https://aclanthology.org/C18-1003 (cited on pages 12, 95, 96).

Michael H. Zhu and Suyog Gupta (2018). *To Prune, or Not to Prune: Exploring the Efficacy of Pruning for Model Compression*. URL: https://openreview.net/forum?id=S1lN69AT- (cited on page 150).

Yilun Zhu, Sameer Pradhan, and Amir Zeldes (Aug. 2021). "OntoGUM: Evaluating Contextualized SOTA Coreference Resolution on 12 More Genres". *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the*

11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). Online: Association for Computational Linguistics, pages 461–467. DOI: 10.18653/v1/2021.acl-short.59. URL: https://aclanthology.org/2021.acl-short.59 (cited on page 12).

Vita

Patrick Xia joined the Center of Language and Speech Processing in the Computer Science Department at Johns Hopkins University in 2016 and received his M.S.E in 2018. His research interests are in information extraction, specifically coreference resolution, efficient methods for NLP, and pretrained language models. He received his Bachelor's degrees in Computer Science and Mathematics from Carnegie Mellon University in 2016.