Moving on from OntoNotes: Coreference Resolution Model Transfer

Patrick Xia
Johns Hopkins University
`paxia@cs.jhu.edu`

Benjamin Van Durme
Johns Hopkins University
`vandurme@cs.jhu.edu`

Abstract

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

1 Introduction

Starting initially with neurally-learned features (Clark and Manning, 2016a,b), end-to-end neural models for coreference resolution (coref) (Lee et al., 2017, 2018) have been developed and imbued with the benefits from contextualized language modeling (Joshi et al., 2019, 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 van 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 coref.

We 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., 2020; Schick and Schütze, 2021). While transfer in other NLP tasks have been studied more in-depth, transfer in coref has scarcely been examined despite recent models containing hundreds of millions of parameters. We 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).[1]

We contribute the first study of neural coref transfer, showing that continued training is effective on eleven datasets spanning different domains, annotation guidelines, and languages. We find evidence that OntoNotes, a widely-used but license-requiring dataset for benchmarking coref, is no better at model transfer than the freely-available PreCo. We establish modern benchmarks on several understudied datasets, including state-of-the-art results on PreCo. Additionally, we analyze practical considerations regarding model selection, catastrophic forgetting, and parameter sharing.[2]

2 Coreference Resolution

Entity coreference resolution is the task of finding clusters of mentions within a document that all refer to the same entity. It still remains a difficult challenge in NLP due to several factors like ambiguity (Poesio and Artstein, 2008) and dependence on real-world knowledge (Levesque et al., 2012).

There are several large annotated datasets for coreference resolution. Annotation guidelines for coref differ across these datasets based on the intended goals of the creators, resulting in differ-

---

[1] We 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).

[2] Code and pretrained models are available at `https://nlp.jhu.edu/coref-transfer`.

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 1: 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.

ences in what is considered a mention, how to handle singleton clusters,[3] and what types of links should be annotated. 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 1 shows the differences between OntoNotes and a few other datasets considered in this work.

However, OntoNotes-based models may not always be 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. 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, we aim 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.

3 Methods

In this section, we describe the method, model, datasets, and initialization methods used to investigate the effectiveness of continued training.

3.1 Continued Training

We 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).

---

[3]An entity cluster with only one mention.

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

3.2 Incremental Coreference Model

End-to-end models for coreference resolution broadly have four parts: a text encoder, a scorer for mention *detection*, a scorer for mention pair *linking*, and an algorithm for decoding clusters. The incremental coreference (ICOREF) model (Xia et al., 2020) used in this work is a constant-memory adaptation of the end-to-end neural coreference resolution model (Lee et al., 2017) with improvements from subsequent work that incorporates stronger encoders (Joshi et al., 2019, 2020). By creating explicit clusters and performing mention-cluster linking instead of mention-pair linking, ICOREF naturally produces clusters from linking scores. This memory-efficient model is conceptually similar to other recent cluster-based models (Toshniwal et al., 2020; Yu et al., 2020b). 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 designed around OntoNotes. As a result, we make minor modifications for compatibility with other datasets by ignoring *genre*-specific embeddings and implementing an auxiliary objective for entity mention detection, similar to the one adopted by Zhang et al. (2018). For completion, we reformulate the ICOREF model to more precisely describe these minor changes in Appendix A.

3.3 Data

We explore a total of two source datasets and eleven target datasets, described in Table 2. 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 we refer 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, we note that 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, we refer to the official "dev" set as our test set, and use a separate 500 training examples as our "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[4] 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[5] 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. We only use 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 PTB (Poesio et al., 2018). Thus, this dataset over-

---

[4]Quiz bowl is a trivia competition where passages give increasingly easier hints towards a common answer, such as a book title, author, location, etc.
[5]LDC2013T22

laps with OntoNotes, which also includes sections of PTB. However, we can use ARRAU to study *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 ($^{\text{es}}$), Catalan ($^{\text{ca}}$), Italian ($^{\text{it}}$), and Dutch ($^{\text{nl}}$). Due to dataset overlaps and licensing, we only use the latter four languages in this paper.

3.4 Source models

ICOREF has three trained components: an encoder, a mention scorer, and a mention linker. We explore initializing the encoder only and the full model.

Pretrained encoders For these models, we initialize only the encoder with a pretrained one and randomly initialize the rest of the model. Joshi et al. (2020) trained 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. We name this finetuned encoder SPANBERT-ON. Conneau et al. (2020) trained XLM-R, a cross-lingual encoder, on webcrawled text in 100 languages. It is effective at cross-lingual transfer, including coreference linking (Xia et al., 2021). We use the "large" size of each model, except for one experiment with the "base" size of SPANBERT-ON.

Trained models Alternatively, we can initialize with the full model. TRANSFER (ON) is a model downloaded directly from Xia et al. (2020). We also train models on PreCo with SpanBERT-large (TRANSFER (PC)) and on OntoNotes$^{\text{en}}$ with XLM-R (TRANSFER (EN)).[6] We also train a variant of each model with gold mention boundaries, which skips the mention scorer.

---

[6] We train the cross-lingual models separately because XLM-R and SpanBERT use different tokenization.

4 Experiments and Results

For a single source model and target dataset, we train several models using a different number of input training examples. The exact details for training set sizes and preprocessing are in Appendix B while training details and hardware are in Appendix C. We evaluate coreference using the average F_1 between MUC, B^3 and CEAF$_{\phi_4}$, following prior work (Pradhan et al., 2012).[7]

4.1 How effective is continued training for domain adaptation?

Continued training Figure 1 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, 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. (2020) 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. We corroborate their findings for a document-level information extraction task, since our models, based on strong pretrained encoders, perform well with just 5 or 10 training documents.

OntoNotes vs. PreCo We find 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

---

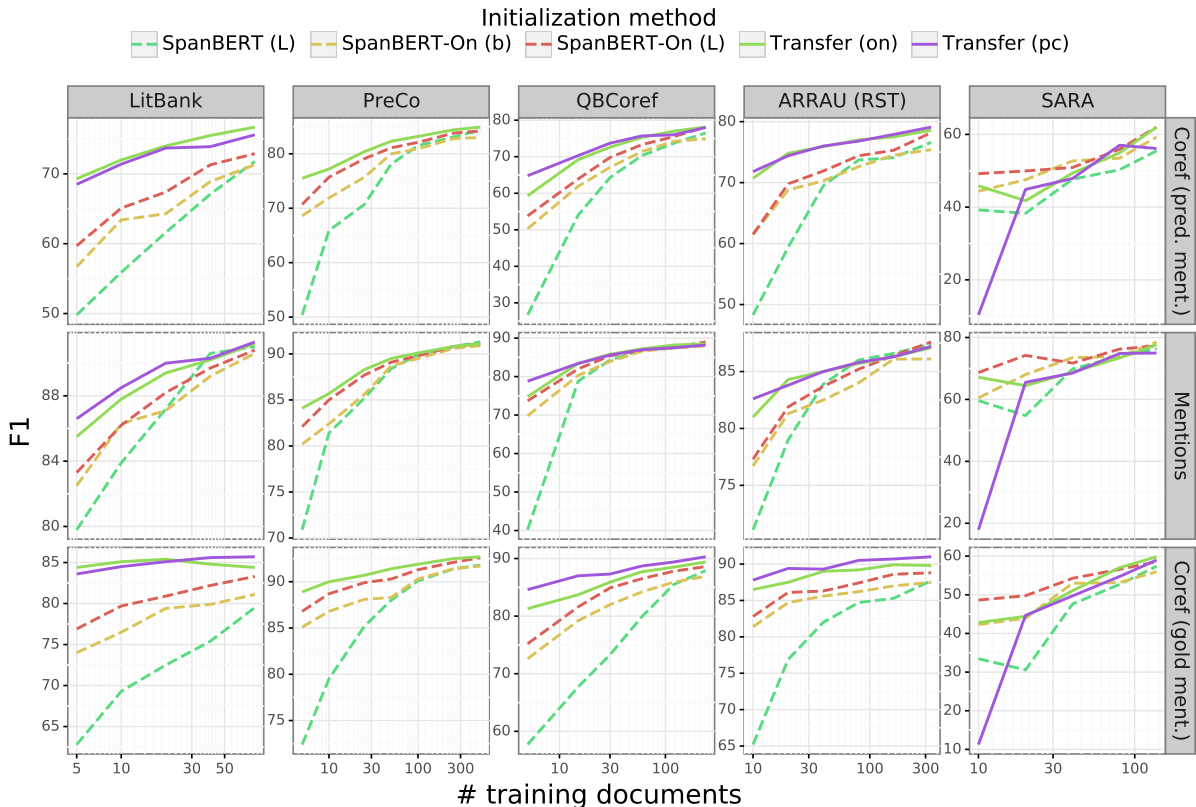[7] We score exact match for SARA (following prior work).

Figure 1: 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).

in low-data settings (e.g. LitBank or QBCoref at 5 examples). However, we subsequently explore the case when all models are given gold mention boundaries in pretraining, continued training, and testing, which would effectively evaluate just the linker. We find in this case that PreCo outperforms OntoNotes even more on QBCoref, LitBank, as well as ARRAU$^{\text{RST}}$. 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, we 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, we see that 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 3 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, we present these as stronger baselines for future work.[8] The purpose is

---

[8]Contemporaneous work has established even stronger baselines for LitBank (Thirukovalluru et al., 2021).

Dataset	Prior work	Previous Model	Previous Score	Our best	Our Model
PreCo	Wu and Gardner (2020)	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 3: Test F_1 on all datasets and the previous state-of-the-art on each dataset, to the best of our knowledge. Again, we are benchmarking the general method of continued training described in this paper, which will not necessarily outperform models that incorporate domain or language specific knowledge. Our best TRANSFER model is determined by the dev set (Appendix C). *ARRAU^{RST} is not directly comparable to prior work as we 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.
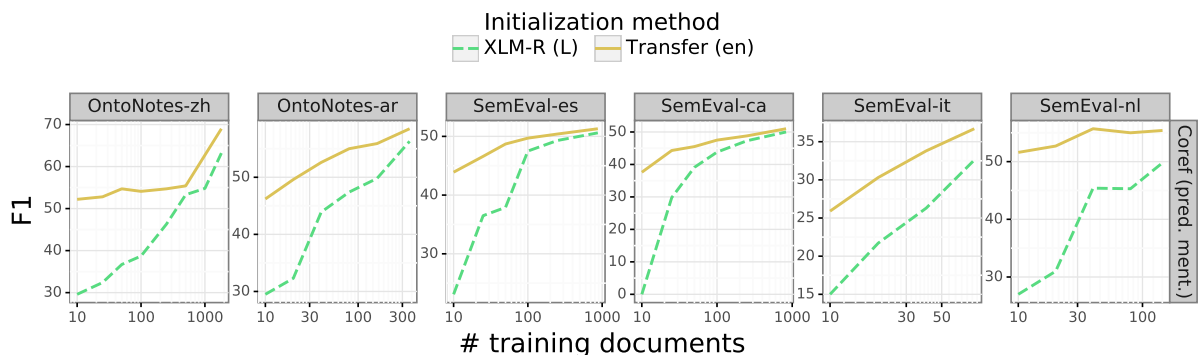

Figure 2: Like Figure 1, this plot demonstates the effectiveness of continued training across different *languages*. XLM-R uses a pretrained encoder (dashed line), while TRANSFER (EN) is first trained on OntoNotes^{en} (solid line). Trends on mention accuracy and using gold mentions look similar and are in Appendix D.

to quantify the effectiveness of continued training and highlight PreCo as an alternative pretraining dataset. Note that we achieve this strong performance without hyperparameter tuning or incorporating any language or domain specific features.

Cross-lingual transfer We present the results for multilingual coreference resolution in Figure 2. 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. Our 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, we expect *joint* multilingual pretraining followed by continued training to be an even more effective recipe in creating the best models for each language. This

is out of scope for this work, which is focused on transfer from single datasets.

4.2 How to allocate annotated documents?

In Figure 1, the experiments for each dataset used the same dev set for model selection to improve comparability. At the same time, we 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. Here, we explore allocating fewer documents for model selection.

We compare 20 models for PreCo trained with a different number of examples using SPANBERT-ON (L) and TRANSFER (ON). We train each model for 60 epochs and make predictions on all 500 dev examples. Next, for each dev set size, we
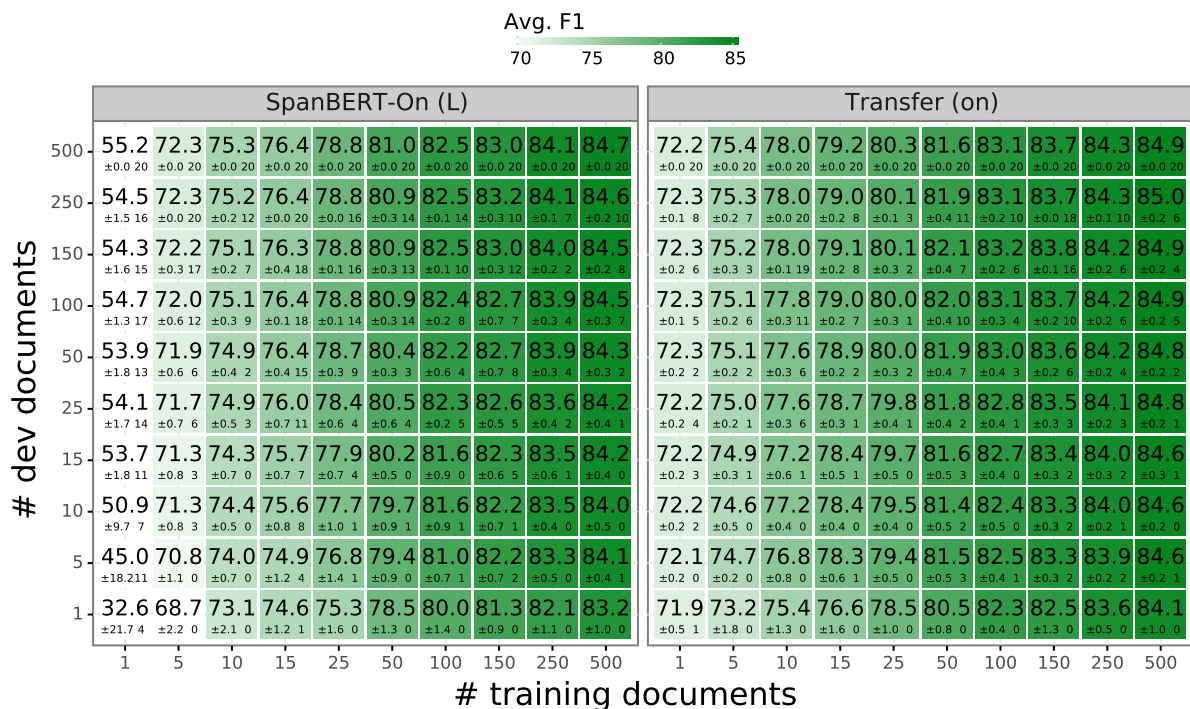
Avg. F1
70 75 80 85

SpanBERT-On (L)

# dev documents	1	5	10	15	25	50	100	150	250	500
500	55.2 ±0.0 20	72.3 ±0.0 20	75.3 ±0.0 20	76.4 ±0.0 20	78.8 ±0.0 20	81.0 ±0.0 20	82.5 ±0.0 20	83.0 ±0.0 20	84.1 ±0.0 20	84.7 ±0.0 20
250	54.5 ±1.5 16	72.3 ±0.0 20	75.2 ±0.2 12	76.4 ±0.0 20	78.8 ±0.0 16	80.9 ±0.3 14	82.5 ±0.1 14	83.2 ±0.3 10	84.1 ±0.1 7	84.6 ±0.2 10
150	54.3 ±1.6 15	72.2 ±0.3 17	75.1 ±0.2 7	76.3 ±0.4 18	78.8 ±0.1 16	80.9 ±0.3 13	82.5 ±0.1 10	83.0 ±0.3 12	84.0 ±0.2 2	84.5 ±0.2 8
100	54.7 ±1.3 17	72.0 ±0.6 12	75.1 ±0.3 9	76.4 ±0.1 18	78.8 ±0.1 14	80.9 ±0.3 14	82.4 ±0.2 8	82.7 ±0.7 7	83.9 ±0.3 4	84.5 ±0.3 7
50	53.9 ±1.8 13	71.9 ±0.6 6	74.9 ±0.4 2	76.4 ±0.4 15	78.7 ±0.3 9	80.4 ±0.3 3	82.2 ±0.6 4	82.7 ±0.7 8	83.9 ±0.3 4	84.3 ±0.3 2
25	54.1 ±1.7 14	71.7 ±0.7 6	74.9 ±0.5 3	76.0 ±0.7 11	78.4 ±0.6 4	80.5 ±0.6 4	82.3 ±0.2 5	82.6 ±0.5 5	83.6 ±0.4 2	84.2 ±0.4 1
15	53.7 ±1.8 11	71.3 ±0.8 3	74.3 ±0.7 0	75.7 ±0.7 7	77.9 ±0.7 4	80.2 ±0.5 0	81.6 ±0.9 0	82.3 ±0.6 1	83.5 ±0.6 1	84.2 ±0.4 0
10	50.9 ±9.7 7	71.3 ±0.8 3	74.4 ±0.5 0	75.6 ±0.8 8	77.7 ±1.0 1	79.7 ±0.9 1	81.6 ±0.9 1	82.3 ±0.7 1	83.5 ±0.4 0	84.0 ±0.5 0
5	45.0 ±18.2 11	70.8 ±1.1 0	74.0 ±0.7 0	74.9 ±1.2 4	76.8 ±1.4 1	79.4 ±0.9 0	81.0 ±0.7 1	82.2 ±0.7 2	83.3 ±0.5 0	84.1 ±0.4 1
1	32.6 ±21.7 4	68.7 ±2.2 0	73.1 ±2.1 0	74.6 ±1.2 1	75.3 ±1.6 0	78.5 ±1.3 0	80.0 ±1.4 0	81.3 ±0.9 0	82.1 ±1.1 0	83.2 ±1.0 0

Transfer (on)

# dev documents	1	5	10	15	25	50	100	150	250	500
500	72.2 ±0.0 20	75.4 ±0.0 20	78.0 ±0.0 20	79.2 ±0.0 20	80.3 ±0.0 20	81.6 ±0.0 20	83.1 ±0.0 20	83.7 ±0.0 20	84.3 ±0.0 20	84.9 ±0.0 20
250	72.3 ±0.1 8	75.3 ±0.2 7	78.0 ±0.0 20	79.0 ±0.2 8	80.1 ±0.1 3	81.9 ±0.4 11	83.1 ±0.2 10	83.7 ±0.0 18	84.3 ±0.1 10	85.0 ±0.2 6
150	72.3 ±0.2 6	75.2 ±0.3 3	78.0 ±0.1 19	79.1 ±0.2 8	80.1 ±0.3 2	82.1 ±0.4 7	83.2 ±0.2 6	83.8 ±0.1 16	84.2 ±0.2 6	84.9 ±0.2 4
100	72.3 ±0.1 5	75.1 ±0.2 6	77.8 ±0.3 11	79.0 ±0.2 7	80.0 ±0.3 1	82.0 ±0.4 10	83.1 ±0.3 4	83.7 ±0.2 10	84.2 ±0.2 6	84.9 ±0.2 5
50	72.3 ±0.2 2	75.1 ±0.2 2	77.6 ±0.3 6	78.9 ±0.2 2	80.0 ±0.3 2	81.9 ±0.4 7	83.0 ±0.4 3	83.6 ±0.2 6	84.2 ±0.2 4	84.8 ±0.2 2
25	72.2 ±0.2 4	75.0 ±0.2 1	77.6 ±0.3 6	78.7 ±0.3 1	79.8 ±0.4 1	81.8 ±0.4 2	82.8 ±0.4 1	83.5 ±0.3 3	84.1 ±0.2 3	84.8 ±0.2 1
15	72.2 ±0.2 3	74.9 ±0.3 1	77.2 ±0.6 1	78.4 ±0.5 1	79.7 ±0.5 0	81.6 ±0.5 3	82.7 ±0.4 0	83.4 ±0.3 2	84.0 ±0.3 1	84.6 ±0.3 1
10	72.2 ±0.2 2	74.6 ±0.5 0	77.2 ±0.4 0	78.4 ±0.4 0	79.5 ±0.4 0	81.4 ±0.5 2	82.4 ±0.5 0	83.3 ±0.2 2	84.0 ±0.2 1	84.6 ±0.2 0
5	72.1 ±0.2 0	74.7 ±0.2 0	76.8 ±0.8 0	78.3 ±0.6 1	79.4 ±0.5 0	81.5 ±0.5 3	82.5 ±0.4 1	83.3 ±0.3 2	83.9 ±0.2 2	84.6 ±0.2 1
1	71.9 ±0.5 1	73.2 ±1.8 0	75.4 ±1.3 0	76.6 ±1.6 0	78.5 ±1.0 0	80.5 ±0.8 0	82.3 ±0.4 0	82.5 ±1.3 0	83.6 ±0.5 0	84.1 ±1.0 0

training documents

Figure 3: 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 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.

sample a subset of the full predictions and determine, post-hoc, the checkpoint at which the model would have stopped had we used that sampled subset. We sample 20 such subsets and compute the expected scores and standard deviation for each model, along with how frequently the subset agreed with the full dev set.

Figure 3 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 beneficial to allocate as many documents as possible towards training, leaving behind a small set for model selection.

4.3 How much do the source models forget?

To measure the degree of catastrophic forgetting (McCloskey and Cohen, 1989), we revisit the source datasets of each TRANSFER model and track its performance in the presence of more training data.[9] In Figure 4, we see that on some datasets, the performance difference is especially pronounced after training on just 10 examples in the target dataset.

We 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, we observe that 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.

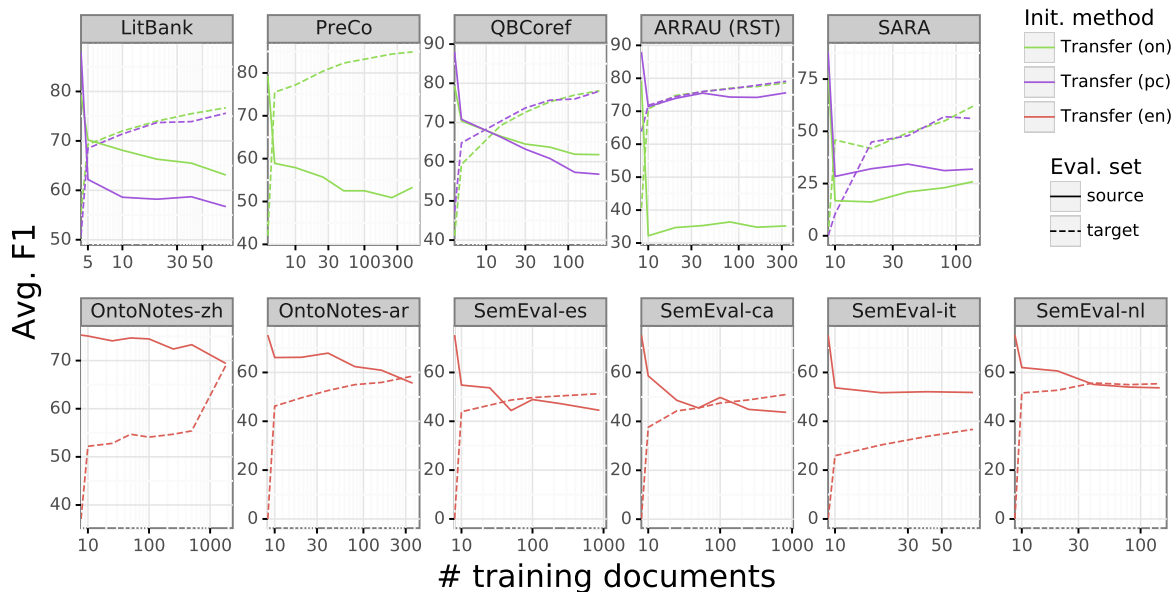[9] For datasets with k-folds, we plot the mean across folds.

Figure 4: 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 1) while the solid lines show performance on the original dataset.
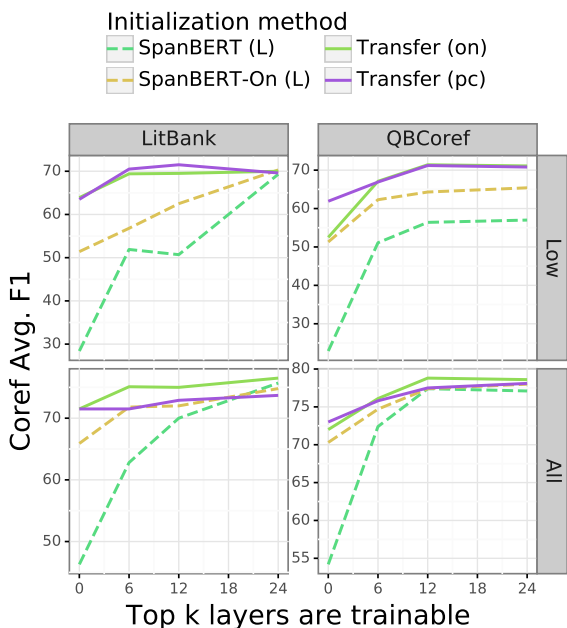

Figure 5: Average F_1 across different models and number of trainable layers. *Low* vs. *All* describes the number of documents used for the first fold of LitBank (10 vs. 80) and QBCoref (15 vs. 240).

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 target dataset. We consider 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. We investigate LitBank and QBCoref under low and high(er) data conditions. This is motivated by 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., 2019).

Figure 5 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, we observe that 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 observed in both datasets and data quantities.[10]

This demonstrates that continued training allows us to freeze a substantial fraction of the model and still achieve good performance. In a multi-dataset scenario, this would also reduce the total number of parameters as the lower layers of the encoder can be shared. This is impactful for neural coref models because recent improvements are due to encoders that are also growing in size (340M for SpanBERT (L) and 559M for XLM-R), which are significantly larger than the rest of the model (40M).

---

[10]This is also observed for OntoNotes^{zh} and in medium data conditions, detailed in Appendix D.

5 Conclusion

We comprehensively examine the transferability of neural coreference resolution models. We 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, we find that PreCo can be effectively used for pretraining, suggesting a viable alternative to OntoNotes.

In our analysis, we 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.

References

Berfin Aktaş, Veronika Solopova, Annalena Kohnert, and Manfred Stede. 2020. Adapting coreference resolution to Twitter conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2454–2460, Online. Association for Computational Linguistics.

Abdulrahman Aloraini, Juntao Yu, and Massimo Poesio. 2020. Neural coreference resolution for Arabic. In *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 99–110, Barcelona, Spain (online). Association for Computational Linguistics.

Giuseppe Attardi, Maria Simi, and Stefano Dei Rossi. 2010. TANL-1: Coreference resolution by parse analysis and similarity clustering. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 108–111, Uppsala, Sweden. Association for Computational Linguistics.

David Bamman, Olivia Lewke, and Anya Mansoor. 2020. An annotated dataset of coreference in English literature. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 44–54, Marseille, France. European Language Resources Association.

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. 2020. Language models are few-shot learners.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Chen Chen and Vincent Ng. 2012. Chinese noun phrase coreference resolution: Insights into the state of the art. In *Proceedings of COLING 2012: Posters*, pages 185–194, Mumbai, India. The COLING 2012 Organizing Committee.

Hong Chen, Zhenhua Fan, Hao Lu, Alan Yuille, and Shu Rong. 2018. PreCo: A large-scale dataset in preschool vocabulary for coreference resolution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 172–181, Brussels, Belgium. Association for Computational Linguistics.

Kevin Clark and Christopher D. Manning. 2016a. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas. Association for Computational Linguistics.

Kevin Clark and Christopher D. Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Berlin, Germany. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco

Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1982, Seattle, Washington, USA. Association for Computational Linguistics.

Xing Fan, Emilio Monti, Lambert Mathias, and Markus Dreyer. 2017. Transfer learning for neural semantic parsing. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 48–56, Vancouver, Canada. Association for Computational Linguistics.

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

Anupam Guha, Mohit Iyyer, Danny Bouman, and Jordan Boyd-Graber. 2015. Removing the training wheels: A coreference dataset that entertains humans and challenges computers. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1108–1118, Denver, Colorado. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

G. E. Hinton and R. R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Nils Holzenberger and Benjamin Van Durme. 2021. Factoring statutory reasoning as language understanding challenges. In *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)*, pages 2742–2758, Online. Association for Computational Linguistics.

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

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

Vidur Joshi, Matthew Peters, and Mark Hopkins. 2018. Extending a parser to distant domains using a few dozen partially annotated examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1190–1199, Melbourne, Australia. Association for Computational Linguistics.

Huda Khayrallah, Brian Thompson, Kevin Duh, and Philipp Koehn. 2018. Regularized training objective for continued training for domain adaptation in neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 36–44, Melbourne, Australia. Association for Computational Linguistics.

Hamidreza Kobdani and Hinrich Schütze. 2010. SUCRE: A modular system for coreference resolution. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 92–95, Uppsala, Sweden. Association for Computational Linguistics.

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

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

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561. AAAI Press.

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.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *International Workshop on Spoken Language Translation*.

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

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press.

Nafise Sadat Moosavi and Michael Strube. 2018. Using linguistic features to improve the generalization capability of neural coreference resolvers. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Brussels, Belgium. Association for Computational Linguistics.

Massimo Poesio and Ron Artstein. 2008. Anaphoric annotation in the ARRAU corpus. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Massimo Poesio, Yulia Grishina, Varada Kolhatkar, Nafise Moosavi, Ina Roesiger, Adam Roussel, Fabian Simonjetz, Alexandra Uma, Olga Uryupina, Juntao Yu, and Heike Zinsmeister. 2018. Anaphora resolution with the ARRAU corpus. In *Proceedings of the First Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 11–22, New Orleans, Louisiana. Association for Computational Linguistics.

Corbèn Poot and Andreas van Cranenburgh. 2020. A benchmark of rule-based and neural coreference resolution in Dutch novels and news. In *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 79–90, Barcelona, Spain (online). Association for Computational Linguistics.

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

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501, Cambridge, MA. Association for Computational Linguistics.

Marta Recasens, Lluís Màrquez, Emili Sapena, M. Antònia Martí, Mariona Taulé, Véronique Hoste, Massimo Poesio, and Yannick Versley. 2010. SemEval-2010 task 1: Coreference resolution in multiple languages. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 1–8, Uppsala, Sweden. Association for Computational Linguistics.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 8–14, New Orleans, Louisiana. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Raghuveer Thirukovalluru, Nicholas Monath, Kumar Shridhar, Manzil Zaheer, Mrinmaya Sachan, and Andrew McCallum. 2021. Scaling within document coreference to long texts. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3921–3931, Online. Association for Computational Linguistics.

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. In *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*, volume 2831 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Shubham Toshniwal, Sam Wiseman, Allyson Ettinger, Karen Livescu, and Kevin Gimpel. 2020. Learning to Ignore: Long Document Coreference with

Bounded Memory Neural Networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8519–8526, Online. Association for Computational Linguistics.

Gorka Urbizu, Ander Soraluze, and Olatz Arregi. 2020. Sequence to sequence coreference resolution. In *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 39–46, Barcelona, Spain (online). Association for Computational Linguistics.

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.

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

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*.

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

Zhaofeng Wu and Matt Gardner. 2020. Understanding mention detector-linker interaction for neural coreference resolution.

Patrick Xia, Guanghui Qin, Siddharth Vashishtha, Yunmo Chen, Tongfei Chen, Chandler May, Craig Harman, Kyle Rawlins, Aaron Steven White, and Benjamin Van Durme. 2021. LOME: Large ontology multilingual extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 149–159, Online. Association for Computational Linguistics.

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

Haoran Xu, Seth Ebner, Mahsa Yarmohammadi, Aaron Steven White, Benjamin Van Durme, and Kenton Murray. 2021. Gradual fine-tuning for low-resource domain adaptation. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 214–221, Kyiv, Ukraine. Association for Computational Linguistics.

Jian Bo Yang, Qi Mao, Qiao Liang Xiang, Ivor Wai-Hung Tsang, Kian Ming Adam Chai, and Hai Leong Chieu. 2012. Domain adaptation for coreference resolution: An adaptive ensemble approach. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 744–753, Jeju Island, Korea. Association for Computational Linguistics.

Juntao Yu, Nafise Sadat Moosavi, Silviu Paun, and Massimo Poesio. 2020a. Free the plural: Unrestricted split-antecedent anaphora resolution. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6113–6125, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Juntao Yu, Nafise Sadat Moosavi, Silviu Paun, and Massimo Poesio. 2021. Stay together: A system for single and split-antecedent anaphora resolution. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4174–4184, Online. Association for Computational Linguistics.

Juntao Yu, Alexandra Uma, and Massimo Poesio. 2020b. A cluster ranking model for full anaphora resolution. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 11–20, Marseille, France. European Language Resources Association.

Rui Zhang, Cícero Nogueira dos Santos, Michihiro Yasunaga, Bing Xiang, and Dragomir Radev. 2018. Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 102–107, Melbourne, Australia. Association for Computational Linguistics.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender bias in contextualized word embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 629–634, Minneapolis, Minnesota. Association for Computational Linguistics.

Shanheng Zhao and Hwee Tou Ng. 2014. Domain adaptation with active learning for coreference resolution. In *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi)*, pages 21–29, Gothenburg, Sweden. Association for Computational Linguistics.

A Model

A.1 The ICOREF model

The ICOREF model (Xia et al., 2020) uses an incremental algorithm to perform coreference resolution. Given a text segment of length n with (sub)tokens $x_1 \ldots x_n$, the model 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, we learn a span scoring function $s_m(x_i)$ intended to rank the likelihood the given span is a coreference mention. We prune the number of spans considered in the next step to a manageable number spans, kn, for some ratio k.

The incremental algorithm 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

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

In this work, we instead optimize for *all* antecedents of x, $Ant(x)$, instead of the most recent one:

$$-\log \prod_{x_i \in X} \sum_{y_i \in Ant(x)} \frac{1}{|Ant(x_i)|} P(c_{y_i} | x_i). \quad (1)$$

We find that this leads to comparable (or slightly better) performance.

Finally, s_a usually incorporates a genre embedding determined by the genre of the document. We retain that small set of parameters but assume all documents have the same genre. The only model for which this is not the case is the directly downloaded model, as it was trained for best performance on OntoNotes.

For most datasets and many downstream tasks, we want to include the singleton entity mentions in the output predictions. For OntoNotes, all singleton mentions are removed in postprocessing. We could add an auxiliary objective that maximizes $s_m(x_i)$ if x_i is an entity mention (Zhang et al., 2018) and only prune out singleton mentions $s_m(x_i) < 0$ in postprocessing. Instead, we present a model reformulation that is similar to the choices made by Toshniwal et al. (2020).

Instead of taking the top kn spans at span pruning, we prune to the top kn spans from the set $\{x_i \in X : s_m(x_i) > 0\}$ (which could have fewer than kn elements). This is both more efficient and easier to optimize for. Now, the training objective is to minimize $s_m(x_i)$ if x_i is not an entity mention, and maximize $s_m(x_i) + s_a(x_i, c_j)$ if it is. This latter term is identical to $s_c(x_i)$ from the previous model.

We can interpret this change 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 would best match to (s_a). We can decompose the joint probability,

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

This can further split into the components,

$$P(E|M=1, x_i) = \frac{\exp(s_a(x_i, c^*_{x_i}))}{\sum_{c_j \in C_\epsilon} \exp(s_a(x_i, c_j))} \quad (2)$$

$$P(E|M=0, x_i) = 1 \quad (3)$$

$$P(M=1, x_i) = \frac{\exp(s_m(x_i))}{1 + \exp s_m(x_i)} \quad (4)$$

$$P(M=0, x_i) = 1 - P(M=1, x_i) \quad (5)$$

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 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.

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 we know $M = 0$, then we 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, we found that this makes no difference in performance on the task, though pruning spans earlier resulted in a substantially faster model.

B   Dataset Preprocessing

We use the scripts from Joshi et al. (2019) to convert all documents into sentence-separated and subtokenized segments of sizes at most 512. For all English datasets, we use the SpanBERT tokenizer, while we use the XLM-R tokenizer for the cross-lingual experiments.

For QBCoref, we split the dataset into five splits after shuffling the initial dataset. For LitBank, we use the published splits (Bamman et al., 2020). In ARRAU^{RST}, several mentions are split. Correctly modeling split spans is an active area of ongoing work (Yu et al., 2020a, 2021). Since we use ARRAU^{RST} primarily for intrinsic comparisons, we defer to the *minimum* span if a mention is split. This means we replaced a subset of markables, listed in Table 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.

Table 5 shows the number of training examples we use for each dataset. Since we only shuffle once initially, larger training sets are always a superset of a smaller one.

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: Training set sizes considered for each dataset. * For SARA, we use the entire fold, which contains 138 documents on average.

C   Training Details

We follow the same hyperparameters used by Xia et al. (2020). We use $k = 0.4$ to select the top $0.4n$ spans, use learning rates of 2e-4 for training the non-encoder parameters (with Adam) and 1e-5 for the encoder (with AdamW). For all models, we finetune the full encoder. We use gradient clipping of 10, train for up to 100 epochs with a patience of 10 for early stopping, as determined by dev F1. We consider spans up to 10 for SARA, 15 for PreCo and ARRAU^{RST}, 20 for LitBank and QBCoref, and 30 for all other datasets. These choices are made based on prior work or the statistics of the training set; increasing the value would affect runtime (with marginal gains in performance).

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.

For Table 3, we pick the best model between TRANSFER (ON) and TRANSFER (PC) based on their dev scores on each dataset. These are listed in Table 6

D   Full cross-lingual results

D.1   Continued Training (4.1)

Figure 6 is the full figure, analogous to Figure 1.

D.2   Training Top Layers (4.4)

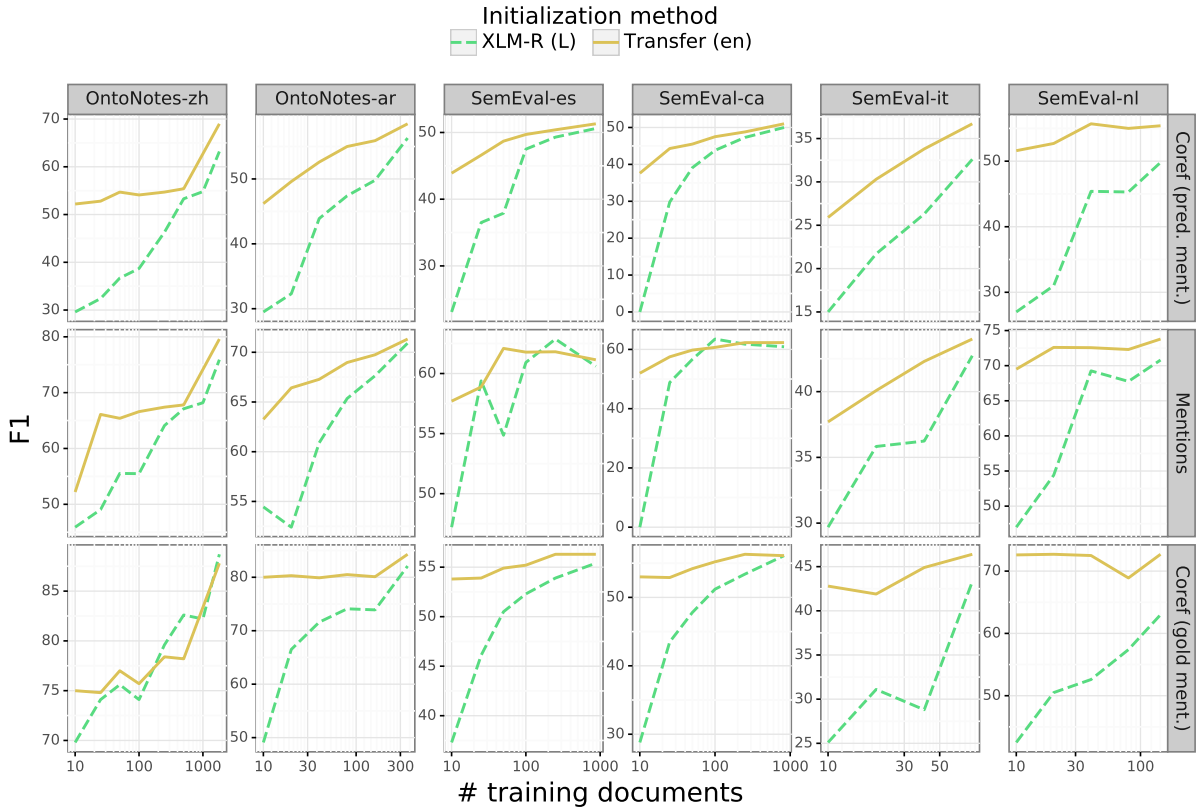For each dataset, we include a "medium" data volume (40 for LitBank, 60 for QBCoref) and we also

Figure 6: Full version of Figure 2. Like Figure 1, 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).

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 6: Dev. F1 scores on each of the models and datasets presented in Table 3. For the English dataset, the test score of the model with the best performing score is reported in Table 3.

include OntoNotes^zh with 50, 500, and 1810 for the three data volumes respectively. These plots are presented in Figure 7. These trends follow what is described in subsection 4.4. Notably, freezing the lower layers when training OntoNotes^zh from scratch appears to consistently outperform training the full model.
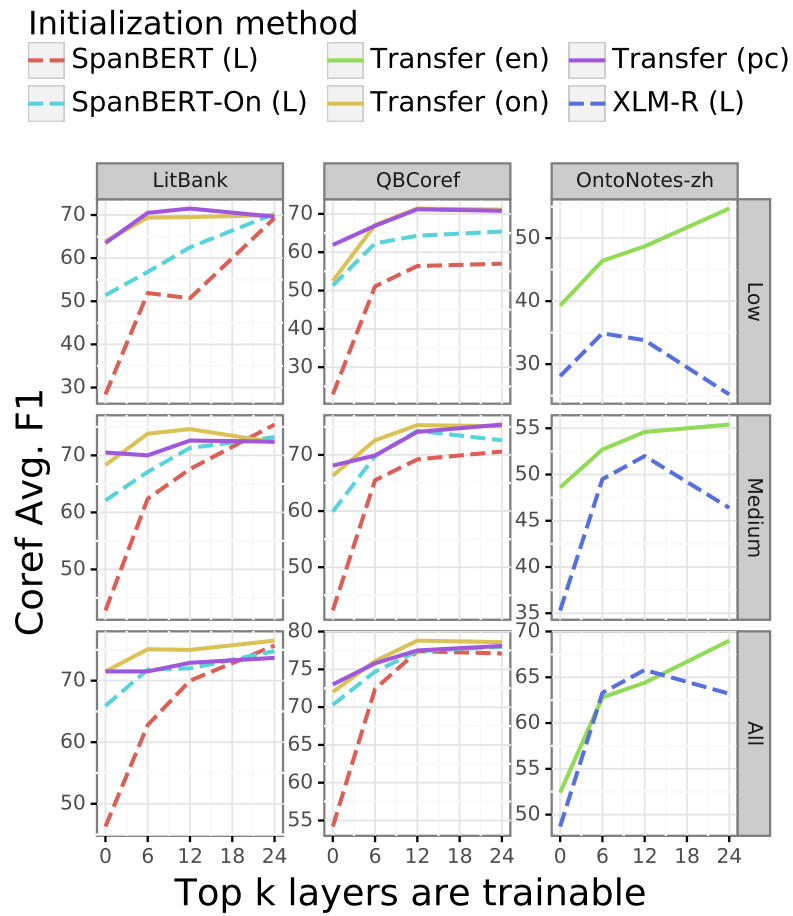
5255

Figure 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.