LongtoNotes: OntoNotes with Longer Coreference Chains

Anonymous ACL submission

Abstract

Ontonotes has served as the most important benchmark for coreference resolution. However, for ease of annotation, several long documents in Ontonotes were split into smaller parts. In this work, we build a corpus of coreference-annotated documents of significantly longer length than what is currently available. We do so by providing an accurate, manually-curated, merging of annotations from documents that were split into multiple parts in the original Ontonotes annotation process (Pradhan et al., 2013). The resulting corpus, which we call LongtoNotes contains documents in multiple genres of the English language with varying lengths, the longest of which are up to 8x the length of documents in Ontonotes, and 2x those in Litbank. We evaluate state-of-the-art neural coreference systems on this new corpus, analyze the relationships between model architectures/hyperparameters and document length on performance and efficiency of the models, and demonstrate areas of improvement in long-document coreference modelling revealed by our new corpus.

1 Introduction

Coreference resolution is an important problem in discourse with applications in knowledge-base construction (Luan et al., 2018), question-answering (Reddy et al., 2019) and reading assistants (Azab et al., 2013; Head et al., 2021). In many such settings, the documents of interest, are significantly longer and/or on wider varieties of domains than the currently available corpora with coreference annotation (Pradhan et al., 2013; Bamman et al., 2019; Mohan and Li, 2019; Cohen et al., 2017).

The Ontonotes corpus (Pradhan et al., 2013) is perhaps the most widely used benchmark for coreference (Lee et al., 2013a; Durrett and Klein, 2013; Wiseman et al., 2016; Lee et al., 2017; Joshi et al., 2020; Toshniwal et al., 2020b; Thirukovalluru et al., 2021; Kirstain et al., 2021). The construction process for Ontonotes, however, resulted in documents

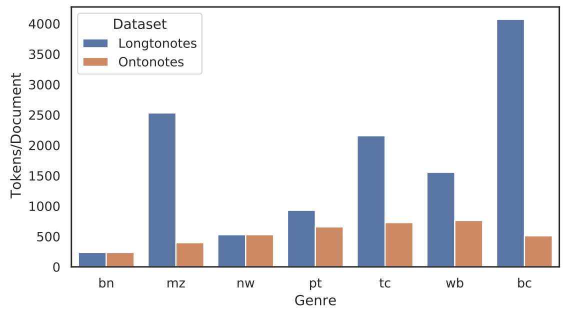


Figure 1: **Comparing Average Document Length.** Long documents in genres such as *broadcast conversations (bc)* were split into smaller parts in Ontonotes. Our proposed dataset, `LongtoNotes`, restores documents to their original form, revealing dramatic increases in length in certain genres.

with an artificially reduced length. For ease of annotation, longer documents were split into smaller parts and each part was annotated separately and treated as an independent document (Pradhan et al., 2013). The result is a corpus in which certain genres, such as *broadcast conversation (bc)*, have greatly reduced length compared to their original form (Figure 1). As a result, the long, bursty spread of coreference chains in these documents is missing from the evaluation benchmark.

In this work, we present an extension to the Ontonotes corpus, called `LongtoNotes`. `LongtoNotes` combines coreference annotations in various parts of the same document, leading to a full document coreference annotation. This was done by our annotation team, which was carefully trained to follow the annotation guidelines laid out in the original Ontonotes corpus (§3). This led to a dataset where the average document length is over 40% longer than the standard OntoNotes benchmark and the average size of coreference chains increased by 25%. While other datasets such as Litbank (Bamman et al., 2019) and CRAFT (Cohen et al., 2017) focus on long documents in specialized domains, `LongtoNotes` comprises of documents in multiple genres (Table 1).

To illustrate the usefulness of LongtoNotes, we evaluate state-of-the-art coreference resolution models (Kirstain et al., 2021; Toshniwal et al., 2020b; Joshi et al., 2020) on the corpus and analyze the performance in terms of document length (§4.2). We illustrate how model architecture decisions and hyperparameters that support long-range dependencies have the greatest impact on coreference performance and importantly, these differences are only illustrated using LongtoNotes and are not seen in Ontonotes (§4.3). LongtoNotes also presents a challenge in scaling coreference models as prediction time and memory requirement increase substantially on the long documents (§4.4).

2 Our Contribution: LongtoNotes

We present LongtoNotes, a corpus that extends the English coreference annotation in the OntoNotes Release 5.0 corpus[1] (Pradhan et al., 2013) to provide annotations for longer documents. In the original English OntoNotes corpus, the genres such as *broadcast conversations (bc)* and *telephone conversation (tc)* contain long documents that were divided into smaller parts to facilitate easier annotation. LongtoNotes is constructed by collecting annotations to combine within-part coreference chains into coreference chains over the entire long document. The annotation procedure, in which annotators merge coreference chains, is described and analyzed in Section 3.

The divided parts of a long document in Ontonotes are all assigned to the same partition (train/dev/test). This allows LongtoNotes to maintain the same train/dev/test partition, at the document level, as Ontonotes (Appendix, Table 10). The size of these partitions however does change as the divided parts are combined into a single annotated text in LongtoNotes. We will release scripts to convert OntoNotes to LongtoNotes in both CoNLL and CorefUD (Universal Dependencies)[2] formats under the Creative Commons 4.0 license . We refer to LongtoNotes$_s$ as the subset of LongtoNotes comprising only of long documents (i.e. documents merged by the annotators).

2.1 Length of Documents in LongtoNotes

The average number of tokens per document (rounded to the nearest integer) in LongtoNotes is 674, 44% higher than in Ontonotes (466). Table 1 breaks down the changes in document length by genre. We observe that the genre with the longest documents is *broadcast conversation* with 4071 tokens per document, which is a dramatic increase from the length of the divided parts in Ontonotes which had 511 tokens per document in the same. The number of coreference chains and the number of mentions per chain grows as well. The long documents that were split into multiple parts during the original OntoNotes annotation are not evenly distributed among the genres of text present in the corpus. In particular, text categories *broadcast news (bn)* and *newswire (nw)* consist exclusively of short non-split documents, which were not affected by the LongtoNotes merging process. A detailed distribution of what documents are merged in LongtoNotes is provided in Table 9 in the Appendix.

2.2 Number of Coreference Chains

As a consequence of the increase in document length, LongtoNotes presents a higher number of coreference chains per document (16), compared to OntoNotes (12). Figure 2 shows the length and number of coreference chains for each document in the two corpora. As expected, the number of chains in a document tends to get larger as the document size increases. For genres with longer average document lengths like *broadcast conversation (bc)*, the increase in the number of chains is as high as 85%, while this increase is only 25% for *pivot (pt)* genre when the document length is comparatively shorter. It is worth noting that the majority of documents had a number of chains in the range of 20 to 50 and only about 20 documents out of 3493 in the OntoNotes dataset had >50 chains per document. For LongtoNotes the number increases

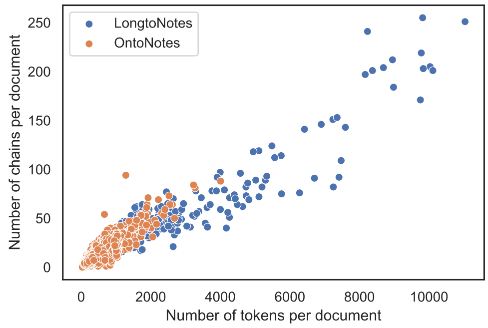


Figure 2: **Document and Coref Chain Length.** The number of coreference chains increases with the increase in token length in LongtoNotes.

[1] The Arabic and Chinese parts of the Ontonotes dataset are not considered in our study. See Appendix A.3

[2] https://ufal.mff.cuni.cz/corefud

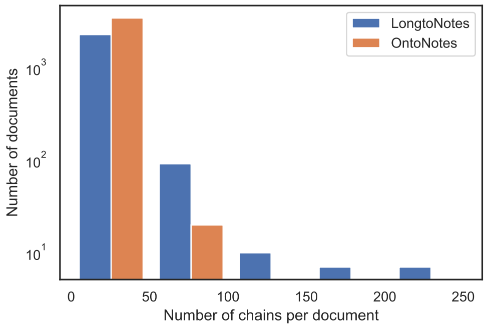


Figure 3: **Number of Chains per Document.** A histogram log plot reveals the long tailed distribution of the number of coreference chains present per document in `LongtoNotes`. Ontonotes contains more documents with fewer chains.

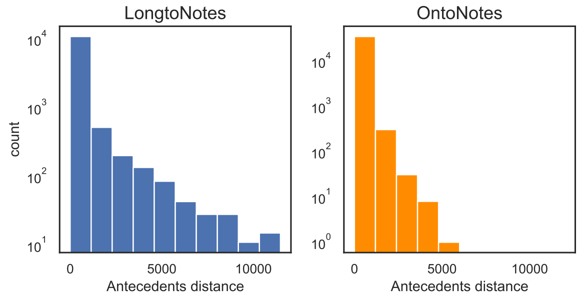


Figure 4: **Distance to Antecedent**. Histogram (log-scale) shows that the largest distance of mention to their antecedents per chain increases in `LongtoNotes` compared to OntoNotes.

to 96 documents. A comparison of the number of chains per document between OntoNotes and `LongtoNotes` is shown in Figure 3.

2.3 Number of Mentions per Chain

The number of mentions per coreference chain in `LongtoNotes` is over 30% more than OntoNotes. This is primarily because of longer documents and an increase in the number of coreference chains per document. Mentions per chain increase with the increase in document length. For the *broadcast conversation (bc)* genre, the increase in the mentions per chain is highest with 87%, while for the *pivot (pt)* (Old Testament and New Testament text) genre it is only 30% as it has shorter documents.

2.4 Distances to the Antecedents

For each coreference chain, we analyzed the distance between the mentions and their antecedents. The largest distance for a mention to its antecedent grew 3x for `LongtoNotes` dataset when compared to OntoNotes from 4,885 to 11,473 tokens. Figure 4 shows a detailed breakdown of the mention to antecedent distance. There are no mentions that are more than 5K tokens distant from its antecedent in OntoNotes. There are 178 such mentions in `LongtoNotes`.

2.5 Comparison with other Datasets

The literature contains multiple works proposing datasets for coreference resolution: Wiki coref (Ghaddar and Langlais, 2016), LitBank (Bamman et al., 2019), PreCo (Chen et al., 2018), Quiz Bowl Questions (Rodriguez et al., 2019; Guha et al., 2015), ACE corpus (Walker et al., 2006), MUC (Chinchor and Sundheim, 1995), MedMentions (Mohan and Li, 2019), inter alia. We compare `LongtoNotes` to these datasets in terms of number of documents, total number of tokens, and document length (Table 2).

Litbank is a popular long document coreference dataset, presenting a high tokens/document ratio. However, the dataset consists of only 100 documents, rendering model development challenges. Moreover, it focuses only on the literary domain. Other datasets containing long documents (e.g., WikiCoref) are also very small in size. On the other hand, datasets consisting of a larger number of texts tend to contain shorter documents (e.g., PreCo). Thus, by building `LongtoNotes` , we address the scarcity of a multi-genre corpus with a collection of long documents containing long-range coreference dependencies.

3 Annotation Procedure & Quality

In this section, we describe and assess the annotation procedure used to build `LongtoNotes`.

3.1 Annotation Task

To build `LongtoNotes` , it suffices to successively merge chains in the current part $i + 1$ of the document with one of the chains in the previous parts $1, \ldots, i$. We reformulate this annotation process as a question answering task where we ask annotators a series of questions (rather the same coreference determining question for different mentions) using our own annotation tool designed for this task (Appendix, Figure 7). We display parts $1, \ldots, i$ with color-coded mention spans. We then show a highlighted concept (a coreference chain in part $i + 1$) and ask the question: *The highlighted*

Categories	# Docs		Tokens/Doc		# Chains		Ment./Chains	
	Ont.	Long.	Ont.	Long.	Ont.	Long.	Ont.	Long.
broadcast conversation (bc)	397	50	511	4071	14	85	65	519
broadcast news (bn)	947	947	237	237	8	8	29	29
magazine (mz)	494	78	398	2531	8	41	32	208
newswire (nw)	922	922	529	529	12	12	47	47
pivot (pt)	369	261	657	930	20	27	131	186
telephone conversation (tc)	142	48	728	2157	17	44	108	319
web data (wb)	222	109	763	1555	17	31	73	149
Overall	3493	2415	466	674	12	16	55	80

Table 1: **Genre Comparison**. Comparison of document and coreference chain statistics per genre in OntoNotes 5.0 and our proposed dataset, `LongtoNotes`.

Dataset	# Docs	Total Size	Tokens/Doc
WikiCoref	30	60K	2000
ACE-2007	599	300K	500
MUC-6	60	30K	500
MUC-7	50	25K	500
QuizBowl	400	50K	125
PreCo	37.6K	12.4M	330
LitBank	100	200K	2105
MedMentions	4392	1.1M	267
OntoNotes	3493	1.6M	466
`LongtoNotes`	2415	1.6M	674
$\texttt{LongtoNotes}_s$	283	740K	2615

Table 2: **Coreference Datasets**. A comparison of various coref datasets with our proposed dataset `LongtoNotes`.

concept below refers to which concept in the above paragraphs?. The annotators select one of the colour-coded chains from parts $1, \ldots, i$ from a list of answers or the annotators can specify that the highlighted concept in part $i + 1$ does not refer to any concept in parts $1, \ldots, i$, (i.e., a new chain emerging in part $i + 1$).

The annotation tool proceeds with a question for each coreference chain ordered (sorted by the first token offset of the first mention in the chain). The annotation of all parts of a document comprises an annotation task. That is, a single annotator is tasked with answering the multiple-choice question for each coreference chain in each part of a document. At the end of each part, annotators are shown a summary page that allows them to review, modify, and confirm the decisions made in the considered part. A screenshot of the summary page is provided in the Fig. 8 in the Appendix.

From Annotations to Coreference Labels The annotations collected in this way are then converted into coreference labels for the merged parts of a document. The answers to the questions tell us the antecedent link between two coreference chains. These links are used to relabel all mentions in the two chains with the same coreference label, resulting in the `LongtoNotes` dataset.

Annotation of singletons Note that the existing OntoNotes coreference annotation does not include singletons. However, considering all parts of a document together might allow mentions that were considered to be singletons in a specific part to be assigned to a coreference chain. To understand the frequency of singletons in a single part of a document that has coreferent mentions in other parts, we manually analysed 500 mentions spread across 10 parts over three randomly selected long documents. We found only 17 instances ($\sim 0.03\%$) where singletons can be merged with coreference chains in different parts of the same document. Given that such singletons would constitute only such a small percentage of mentions, we decided it was appropriate to leave them out of the annotation process to reduce the complexity of the annotation task. To merge this small amount of singleton mentions, our annotators would have had to label over 50% more mentions per document. We further discuss this in Appendix A.4.

3.2 Annotators and Training

We hired and trained a team of three annotators for the aforementioned task. The annotators were university-level English majors from India and were closely supervised by an expert with experience in similar annotation projects. The annotation team was paid a fair wage of approximately 15 USD per hour for the work. We had several hour-long training sessions outlining the annotation task, setup of the problem, and Ontonotes annotation guidelines. We reviewed example cases of difficult annotation decisions and collaboratively worked through example annotations. We then ran a pilot annotation study with a small number of documents (approx 5% of the total documents). For these documents, we also provided annotations to ensure the training of the annotators and eventual annotation quality. We calculated the inter-annotators' agreement between the annotators and us. After a few rounds of training, we were able to achieve an inter-annotator agreement score (strict match, defined in the next subsection) of over 95% and we decided to go ahead with the annotation task. This confirmed the annotators' understanding of the task.

After the satisfactory pilot annotation study, the tasks were assigned to the annotators in five batches of 60 documents each. For 10% of the tasks, we had all three annotators provide annotations. For the remaining 90%, a single annotator was used. For the documents with multiple annotators, we used majority voting to settle disagreements. If all annotators disagreed on a specific case, we selected Annotator 1's decision over the others (analysis in the Appendix).

Did we need annotation? Can the chains be merged automatically? To show the importance of our human-based annotation process, we investigate whether the annotators' decisions could have been replicated using off-the-shelf automatic tools. We performed two experiments: (i) a simple greedy rule-based string matching system (described in the Appendix A.5) and (ii) Stanford rule-based coreference system to merge chains across various parts. We use the merged chains to calculate the CoNLL F_1 score with the annotations produced by our annotators. We found that our string-matching system achieved a CoNLL F_1 score of only 61%, while the Stanford coreference system reached a score of only 69%. The low scores compared to the annotators' agreement (which is over 90%) underline the complexity of the task and the need for such a human-annotated dataset.

3.3 Measuring Quality of Annotation

We would like to ensure that `LongtoNotes` maintains the high-quality standards of OntoNotes. Thus, we compute various metrics of agreement between a pair of annotators. We consider (1) the question-answering agreement (i.e., how similar are the annotations made using the annotation tool), and (2) the coreference label agreement (i.e., at the level of the resulting coreference annotation).

Assume that each annotator receives a set of chains $C_1, C_2, ..., C_N$. For each chain C_i, the annotator links it to a `New` *chain or a chain from their (annotator specific) set of available chains. Let us call D_i this linking decision, which consists of a pair (C_i, A_i), where A_i is the selected antecedent chain.* We consider the following question answering metrics:

(i) Strict Decision Matching: When two annotators agreed on merging two chains and there is an exact match between the merged chains. Calculated as $\frac{1}{N}\sum_i D_i^{(1)} = D_i^{(2)}$.

(ii) Jaccard Decision Match: Jaccard decision calculated as $\frac{1}{N}\sum_i \frac{(D_i^{(1)}.A_i^{(1)}) \cap (D_i^{(2)}.A_i^{(2)})}{(D_i^{(1)}.A_i^{(1)}) \cup (D_i^{(2)}.A_i^{(2)})}$

(iii) New Chain Agreement: Number of times two annotators agreed on a new chain choice divided by the number of times at least one annotator labels *New* chain.

(iv) Not New Chain Agreement: Pairwise agreement between annotators when the chain choice is not a *New* chain.

(v) Krippendorff's alpha: Krippendorff's alpha (Krippendorff, 2011) is the reliability coefficient measuring inter annotator agreement. We compute Krippendorff's alpha using a strict decision match as the coding for agreement.

Metric	Score
Strict Match	0.90
Jaccard Match	0.95
New Chain	0.88
Not New Chain	0.87
Krippendorff's alpha	0.90

Table 3: **Annotation Quality Assessment**. We report the average of each metric over all pairs of annotators.

Table 3 presents the results for these metrics. We observed that on average annotators agreed with each other on over 90% of their decisions except when the *No New* chains were considered. Removing *New* chains reduces the total decisions to be made significantly, and hence a lower score on *No New* chains agreement. We found that Annotator 1 agreed most with the experts and hence Annotator 1's decisions were preferred over the others in case of disagreement between all three annotators.

Where are disagreements found in annotation? We would like to understand what kinds of mentions lead to the disagreement between annotators. To investigate this, we measure the part of speech of all the disagreed chain assignments between the annotators. We found that the 8% of the mentions within the disagreed chain assignments were pronouns, 8% were verbs, and 9% were common nouns. The number of proper nouns disagreements was lower with just 5%. When considering different genres, it was observed that genres with longer documents like *broadcast conversation (bc)* had more mentions that were pronouns when compared with genres with shorter documents *pivot (pt)*. As expected, the number of disagreements in general increased with the size of the documents. However, we found that the number of disagreements was manageably small even for long document genres such as *broadcast conversation (bc)* A more comprehensive overlook is presented in the Appendix.

3.4 Time Taken per Annotation

We also recorded the time taken for each annotation. Time taken per annotation increases with the increase in the document length (Appendix Fig. 9). This is expected as more chains create more options to be chosen from and longer document length demands more reading and attention. In total, our annotation process took 400 hours.

4 Empirical Analysis with `LongtoNotes`

We hope to show that `LongtoNotes` can facilitate the empirical analysis of coreference models in ways that were not possible with the original OntoNotes. We are interested in the following empirical questions using the datasets–Ontonotes (Pradhan et al., 2013), and our proposed `LongtoNotes` and `LongtoNotes`$_s$:

- How does the length of documents play a role in the empirical performance of models?
- Does the empirical accuracy of models depend on different hyperparameters in `LongtoNotes` and Ontonotes?
- Does `LongtoNotes` reveal properties about the efficiency/scalability of models not present in Ontonotes?

4.1 Models

Much of the recent work on coreference can be organized into three categories: span based representations (Lee et al., 2017; Joshi et al., 2020), token-wise representations (Thirukovalluru et al., 2021; Kirstain et al., 2021) and memory networks / incremental models (Toshniwal et al., 2020b,a). We consider one approach from all three categories.

Span based representation We used the Joshi et al. (2020) implementation of the higher-order coref resolution model (Lee et al., 2018) with SpanBERT. Here, the documents were divided into a non-overlapping segment length of 384 tokens. We used SpanBERT Base as our model due to memory constraints. The number of training sentences was set to 3. We set the maximum top antecedents, $K = 50$. We used Adam (Kingma and Ba, 2014) as our optimiser with a learning rate of $2e^{-4}$.

Token-wise representation We used the LongFormer Large (Beltagy et al., 2020) version of Kirstain et al. (2021) work, as this approach is less memory demanding and it is possible to fit this model in our memory. The max sequence length was set to 384 or 4096. Adam was used as an optimiser with a learning rate of $1e^{-5}$. A dropout (Srivastava et al., 2014) probability of 0.3 was used.

Memory networks We used SpanBERT Large with a sequence length of 512 tokens. Following Toshniwal et al. (2020b), an endpoint-based mention detector was trained first and then was used for coreference resolution. The number of training sentences was set to 5, 10, and 20. The number of memory cells was selected from 20 or 40. All experiments were performed with AutoMemory models with learned memory type.

4.2 Length of Documents & Performance

Impact of Training Corpus We first investigate whether or not training on the longer documents in `LongtoNotes` are needed to achieve state-of-the-art results on the dataset. We compare the performance of models trained on Ontonotes to those trained on `LongtoNotes`. We find that by training on `LongtoNotes`, we can achieve higher CoNLL F1 measures on `LongtoNotes` than training with Ontonotes for each model architecture (Table 5). This suggests that the longer dependencies formed by merging annotations in various parts of documents in OntoNotes are difficult to model when training on short documents.

We find that to achieve accuracy with hyperparameters such as learning rate/warmup size, we need to maintain a number of steps per epoch consistent with Ontonotes when training with `LongtoNotes`. A detailed analysis is presented in the Appendix Section C.

Length Analysis - Number of Tokens We break down the performance of the Span-based model by

# Tokens	Training	CoNLL F1
$\leq$ 2K	Ontonotes	**78.85**
	LongtoNotes	78.25
$>$ 2K	Ontonotes	65.11
	LongtoNotes	**66.20**

Table 4: **Performance and Document Length for Span-based Models.** F_1 score across different document length for SpanBERT Base trained model on OntoNotes and LongtoNotes dataset.

the number of tokens in each document. We compare the performance of the model depending on the training set. Figure 2 shows that the majority of the documents in the OntoNotes dataset falls within a token length of 2000 per document. We create two splits of $\texttt{LongtoNotes}_s$, one having a token length greater than 2000 tokens, the other having a number of tokens smaller than 2000. Table 4 shows that for smaller document length (less than 2000 tokens), the SpanBERT model trained on OntoNotes performed better but the trend reverses for longer documents (more than 2000 tokens), on which the model trained on LongtoNotes outperformed the model trained on OntoNotes by $+1\%$.

Length Analysis - Number of Clusters Table 6 displays the change in F_1 score with the increase in the number of clusters per document. The SpanBERT Base model trained on LongtoNotes outperforms the same model trained on OntoNotes ($+0.6\%$) when the number of clusters is more than 40. Note that, 40 is selected based on the cluster distribution shown in Table 1 with the majority documents in LongtoNotes lying in this range.

4.3 Hyperparameters & Document Length

Each model has a set of hyperparameters that would seemingly lead to variation in performance with respect to document length. We consider the performance of the models on LongtoNotes as a function of these hyperparameters.

Span-based model hyperparameters We consider two hyperparameters: the number of antecedents to use, K and the max number of sentences used in each training example. We found that upon varying K: $10, 25$ and 50, there was only a small difference observed in the results for both the models trained on OntoNotes and LongtoNotes (increasing K led to only minor increases). The result is summarized in Table 7. We could not go beyond $K = 50$ due to our GPU memory limitations. However, going beyond 50 might further help for longer documents. Furthermore, we found that the *number of sentences* parameter used to create training batches does not play a significant role in performance either (Figure 5).

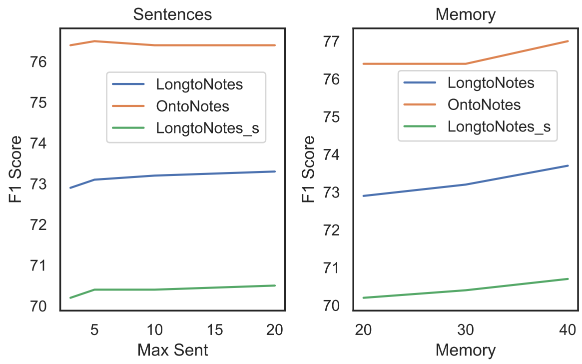


Figure 5: **Max Sentence Length.** Increasing max sentences from 3 to 20 has a small effect on the performance of the SpanBERT large model. On the other hand, the increase is linear with the increase in the memory size alongside the increase in max training sentences.

Token-wise model hyperparameters We experimented with reducing the sequence length when testing from 4096 to 384 and we observe a drop in performance. Figure 6 shows the effect on performance due to the change in the sequence length. We observed that longer sequence length (4096) helps more for $\texttt{LongtoNotes}_s$ as there are longer sequences than for OntoNotes, which is evident in Figure 6. Furthermore, we analyzed the effect of sequence length on two genres: *magazine (mz)* having 6x longer sequences in LongtoNotes than OntoNotes vs *pivot (pt)* having just 1.4x longer documents. As observed in Figure 11, when the document is long as in *magazine (mz)*, there is a significant increase in performance with a longer sequence but the effect is negligible for *pivot (pt)* where the size of the document is almost the same. A detailed comparison is provided in the Appendix Table 15.

Memory model hyperparameters We consider two hyperparameters - the memory size which denotes the maximum active antecedents that can be considered and the max number of sentences used in training. We show that doubling the size of the memory leads to an increase of 0.8 points of CoNLL F_1 for LongtoNotes dataset. (Appendix Table 14). Figure 5 demonstrates that there is no significant improvement in the performance

	Training	OntoNotes P	OntoNotes R	OntoNotes F_1	LongtoNotes$_s$ P	LongtoNotes$_s$ R	LongtoNotes$_s$ F_1	LongtoNotes P	LongtoNotes R	LongtoNotes F_1
Stanford Coref (Lee et al., 2013b)	-	58.6	58.8	58.6	48.5	58.2	52.7	53.6	57.3	55.2
Span-based (Joshi et al., 2020)	OntoNotes	76.5	77.6	**77.4**	72.7	69.1	70.8	74.4	73.0	73.7
	LongtoNotes	75.9	77.7	76.8	72.4	70.7	**71.5**	73.9	74.1	**74.0**
Token-Level (Kirstain et al., 2021)	Ontonotes	81.2	79.5	**80.4**	79.6	80.0	79.8	79.7	77.2	78.5
	LongtoNotes	80.0	78.2	79.1	80.3	80.3	**80.3**	80.2	78.0	**79.1**
Memory-Model (Toshniwal et al., 2020b)	OntoNotes	73.5	79.3	76.4	63.4	73.8	68.2	67.9	76.6	72.0
	LongtoNotes	73.8	79.4	**76.6**	66.3	74.6	**70.2**	69.3	77.0	**72.9**

Table 5: **Performance Variation by Training Set**. Comparison of F_1 scores on various datasets using different models. **All experiments have been performed atleast 2 times and a variance of only** $\pm$ 0.1 **was observed.**

# Chains	Training	SpanBERT	Token	Memory
≤ 40	Onto	**73.60**	**79.80**	**72.80**
	Longto	72.86	78.80	71.94
> 40	Onto	68.44	75.60	67.72
	Longto	**69.09**	**76.42**	**68.60**

Table 6: **Performance and Number of Chains for different models**. CoNLL F_1 score across different document length for SpanBERT Base, Token-Level and Memory-Model trained on OntoNotes and LongtoNotes dataset.

K	OntoNotes	LongtoNotes	LongtoNotes$_s$
10	77.05	73.44	70.37
25	76.93	73.99	**71.61**
50	**77.60**	**74.01**	71.58

Table 7: **Number of Antecedents vs. Performance** SpanBERT Base model trained on LongtoNotes dataset with varying K value.

of the model with the increase in the number of training sentences.

4.4 Model Efficiency

We compare the prediction time for the span-based model on the longest length and average length documents in LongtoNotes and Ontonotes in Table 8. We observe that there is a significant jump in running time and memory required to scale the model to long documents on LongtoNotes; this jump is much smaller on Ontonotes. This suggests that our proposed dataset is better suited for assessing the scaling properties of coreference methods.

5 Conclusion

In this paper, we introduced LongtoNotes, a dataset that merges the coreference annotation of documents that in the original OntoNotes dataset

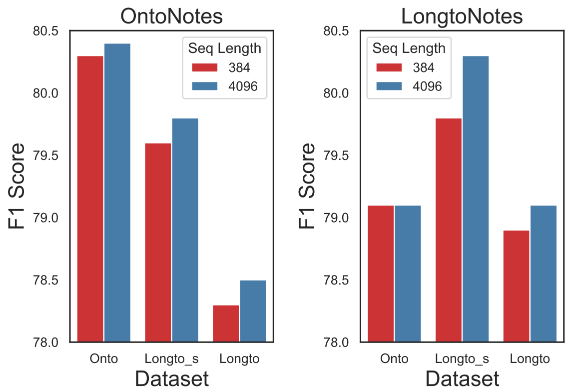


Figure 6: **Sequence Length vs. Performance.** LongFormer is significantly better on LongtoNotes with 4096 sequence length compared to 384. Two sequence lengths perform similarly on Ontonotes.

Dataset	Type	Pred. Time	Pred. Mem
Ontonotes	Average	0.11 sec	1.50 GB
LongtoNotes	Average	0.47 sec	6.50 GB
Ontonotes	Longest	0.37 sec	5.84 GB
LongtoNotes	Longest	2.35 sec	42.68 GB

Table 8: **Model Efficiency of Span-based Models**. We find that LongtoNotes documents have extended length leading to greater variation of prediction time and prediction memory.

were split into multiple independently-annotated parts. LongtoNotes has longer documents and coreference chains than the original OntoNotes dataset. Using LongtoNotes, we demonstrate that scaling current approaches to long documents has significant challenges both in terms of achieving better performance as well as scalability. We demonstrate the merits of using LongtoNotes as an evaluation benchmark for coreference resolution and encourage future work to do so.

Ethical Considerations

Our dataset is comprised solely of English texts, and our analysis, therefore, applies uniquely to the English language. The annotation was performed with a data annotation service which ensured that the annotators were paid fair compensation of 15 USD per hour. The annotation process did not solicit any sensitive information from the annotators. Finally, while our models are not tuned for any specific real-world application, the methods could be used in sensitive contexts such as legal or healthcare settings, and any work building on our methods must undertake extensive quality-assurance and robustness testing before using them.

Replicability: As part of our contributions, we will release the models trained on `LongtoNotes` discussed in this manuscript.

References

Mahmoud Azab, Ahmed Salama, Kemal Oflazer, Hideki Shima, Jun Araki, and Teruko Mitamura. 2013. An NLP-based reading tool for aiding non-native English readers. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 41–48, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566.

David Bamman, Sejal Popat, and Sheng Shen. 2019. An annotated dataset of literary entities. 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 2138–2144, Minneapolis, Minnesota. Association for Computational Linguistics.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Hong Chen, Zhenhua Fan, Hao Lu, Alan L Yuille, and Shu Rong. 2018. Preco: A large-scale dataset in preschool vocabulary for coreference resolution. *arXiv preprint arXiv:1810.09807*.

Nancy A Chinchor and Beth Sundheim. 1995. Message understanding conference (muc) tests of discourse processing. In *Proc. AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 21–26.

K Bretonnel Cohen, Arrick Lanfranchi, Miji Joo-young Choi, Michael Bada, William A Baumgartner, Natalya Panteleyeva, Karin Verspoor, Martha Palmer, and Lawrence E Hunter. 2017. Coreference annotation and resolution in the colorado richly annotated full text (craft) corpus of biomedical journal articles. *BMC bioinformatics*, 18(1):1–14.

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.

Abbas Ghaddar and Phillippe Langlais. 2016. WikiCoref: An English coreference-annotated corpus of Wikipedia articles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 136–142, Portorož, Slovenia. European Language Resources Association (ELRA).

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.

Andrew Head, Kyle Lo, Dongyeop Kang, Raymond Fok, Sam Skjonsberg, Daniel S. Weld, and Marti A. Hearst. 2021. *Augmenting Scientific Papers with Just-in-Time, Position-Sensitive Definitions of Terms and Symbols*. Association for Computing Machinery, New York, NY, USA.

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.

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

Yuval Kirstain, Ori Ram, and Omer Levy. 2021. Coreference resolution without span representations. In *ACL/IJCNLP*.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013a. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational linguistics*, 39(4):885–916.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013b. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.

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. *arXiv preprint arXiv:1804.05392*.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. HLT '05, page 25–32, USA. Association for Computational Linguistics.

Sunil Mohan and Donghui Li. 2019. Medmentions: a large biomedical corpus annotated with umls concepts. *arXiv preprint arXiv:1902.09476*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Pedro Rodriguez, Shi Feng, Mohit Iyyer, He He, and Jordan Boyd-Graber. 2019. Quizbowl: The case for incremental question answering. *arXiv preprint arXiv:1904.04792*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

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.

Shubham Toshniwal, Allyson Ettinger, Kevin Gimpel, and Karen Livescu. 2020a. Petra: A sparsely supervised memory model for people tracking. *arXiv preprint arXiv:2005.02990*.

Shubham Toshniwal, Sam Wiseman, Allyson Ettinger, Karen Livescu, and Kevin Gimpel. 2020b. 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.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Conference on Message Understanding*, MUC6 '95, page 45–52, USA. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57:45.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004, San Diego, California. Association for Computational Linguistics.

Appendix

A Dataset and Annotation Details

A.1 Annotation tool

Figure 7 shows the annotation tool built by us.

A.2 Comparison with OntoNotes

A detailed genre-wise comparison of the documents from OntoNotes dataset which were merged in `LongtoNotes` is presented in Table 9. It can be seen that categories like `bn` and `nw` are completely missing in `LongtoNotes` , while `pt` is partially missing.

Documents in Corpus comparison		
Category	Onto	Longto
bc/cctv	✓	✓
bc/cnn	✓	✓
bc/msnbc	✓	✓
bc/phoenix	✓	✓
bn/abc	✓	✗
bn/cnn	✓	✗
bn/mnb	✓	✗
bn/nbc	✓	✗
bn/pri	✓	✗
bn/voa	✓	✗
mz/sinorama	✓	✓
nw/wsj	✓	✗
nw/xinhua	✓	✗
pt/nt	✓	✓
pt/ot	✓	✗
tc/ch	✓	✓
wb/a2e	✓	✓
wb/c2e	✓	✓
wb/eng	✓	✓

Table 9: **Comparison of documents from various sub-categories that exists in OntoNotes 5.0 and our proposed dataset `LongtoNotes`**

A.3 Dataset selection decision

Due to budget constraints and the expertise of our team and annotators in English only (and some training of annotators is required to ensure data quality), we only considered the English parts of the OntoNotes dataset in our work. We think that the dataset can be extended to Arabic and Chinese too, but we leave it for future work.

A.4 Annotating singletons

While manually annotating all singletons, we observed that almost all NPs can be thought of as mentions and all those NPs that are not part of any chain can be thought of as a singleton. Our analysis suggests that there are over 50% mentions that are not annotated by OntoNotes and can qualify for singletons. To annotate all the singletons, the annotator needs to go through all of them, discard the ones that do not abide by the OntoNotes rules and then make a decision whether to merge each singleton to some chain or other singleton. In our analysis, the number of such singletons is very low and all the efforts were not worth it for the small improvement over the current annotations. So we decide to ignore all the singletons in our study.

A.5 Greedy rule-based matching system

We use a greedy string matching system where we take all the mentions in a chain of the current para $i+1$ and analyse its part of speech provided in the OntoNotes dataset. We take the first Noun (NN or NP) present in each chain and look for the mentions overlap in all other previous paras $1, \ldots, i$ chains. We merged two chains if there is a strict overlap with any of the mentions in a given chain. If there are no strict overlaps, we move to the next noun in the given chain and repeat the process. If we find no strict overlap with any mentions in any other para chains, we keep the chain independent (same as assigning *None of the below* in our annotation tool). We repeat the process with all chains in a given document and constantly update the chain after every para.

B Train test dev split

A comparison between the number of documents in the train-test-dev split between `LongtoNotes` and OntoNotes is provided in Table 10.

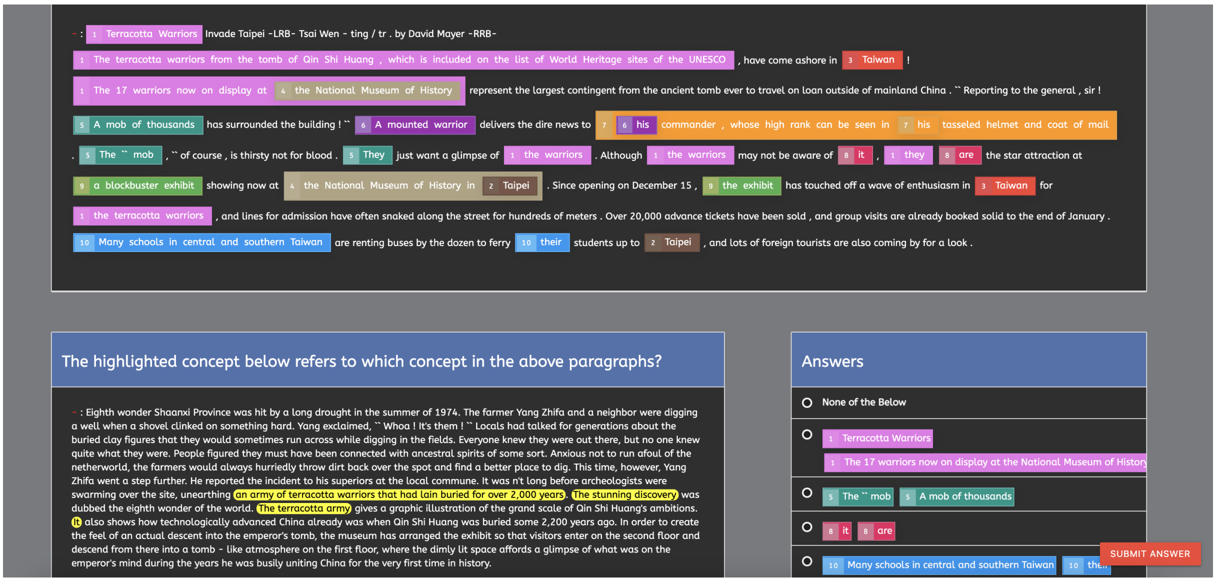
Dataset	Train	Dev	Test
OntoNotes	2802	343	348
`LongtoNotes`	1959	234	222

Table 10: Comparison of the train-test-dev split of documents between OntoNotes and `LongtoNotes`

B.1 Genre wise disagreement analysis

Table 11 presents the genre-wise disagreement analysis for strict decision matching. Genres with longer documents like `bc, mz` have more disagreements compared to genres with smaller document lengths like `tc, pt.`

Figure 7: The tool designed by us for the annotation task. The upper box represents all the previous paragraphs while the box on the bottom left is the current paragraph. The mentions of the current chain to be merged are shown in yellow. On the right side, the answers are presented which are chains from previous paragraphs and the annotator can select one of them or choose the `None of the below` option which creates a new chain.



The trend is very similar for new chain assignments where genres with larger documents have more disagreements over new chain assignments. The numbers are presented in Table 13.

B.2 Annotators disagreements analysis

Figure 10 shows the cases (in black) when the annotators disagreed for each part of the speech categories (shown in big coloured bubbles). The size of the bubbles is representative of their occurrence in the dataset, suggesting there are more pronominal mentions in the dataset than nouns or proper nouns.

B.2.1 Genre wise disagreement analysis

In general, annotators disagree more on pronouns than proper nouns and the trend is consistent for various genres as shown in Table 12.

C Results

C.1 MUC, B^3 and CEAFE scores

Tables 16, 17 and 18 present the MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998) and CEAFE (Luo, 2005) scores for SpanBERT Base (Lee et al., 2017) and LongDocCoref Models (Toshniwal et al., 2020b). On all three metrics, both models trained on `LongtoNotes` dataset outperforms the models trained on OntoNotes dataset. For SpanBERT base model, we compare three version of the `LongtoNotes` dataset: $\texttt{LongtoNotes}_s$ and `LongtoNotes` dataset as mentioned in the paper and $\texttt{LongtoNotes}_{eq}$ where `LongtoNotes` dataset is reweighted to create the total number of documents equal to the number of documents in OntoNotes dataset. For LongDocCoref model, n represents the maximum number of training sentences, while m refers to the memory used.

C.2 Genre wise F_1 scores vs sequence length

Table 15 shows that LongFormer Large model with larger sequence length (4096) outperforms the one with shorter sequence length (384) for all models. The difference is higher when the documents are longer (as seen in `mz` genre) than when the documents are shorter (as seen in `pt`).

Figure 8: The summary page of our annotation tool that is shown after all the chains decisions in a paragraph is made. The annotators can look and verify all the decisions and confirm answers and proceed to the next para or can change their answers if they want.

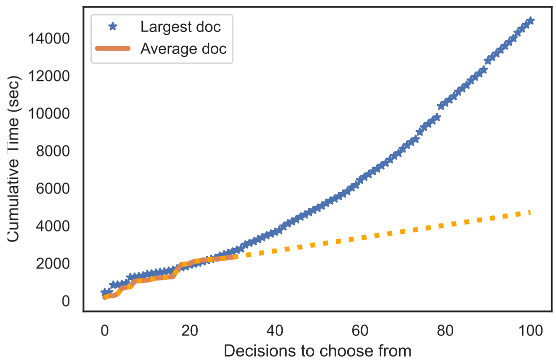


Figure 9: **Annotation Time and Document Length.** Annotation time (cumulative) increases exponentially with the increase in the number of decisions to choose from. A comparison is shown between the longest document in `LongtoNotes` vs an average document. The dotted lines represent the increase in annotation time if the growth was linear.

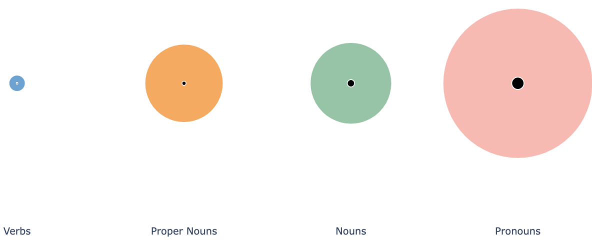


Figure 10: Plot showing the part of speech distribution for the disagreed clusters between annotators.

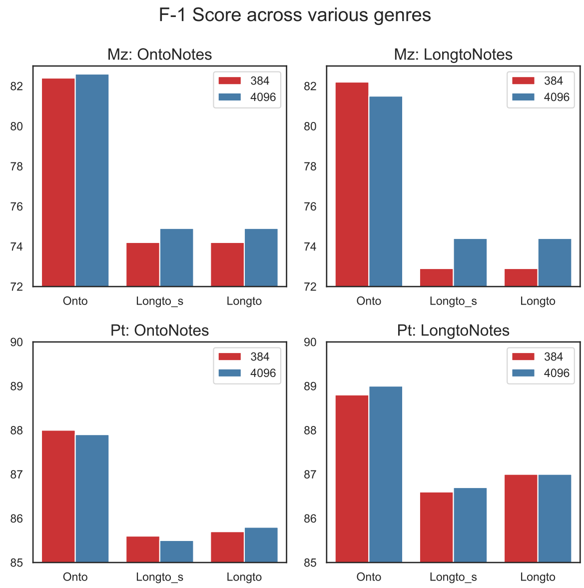


Figure 11: Plot comparing the sequence length effect on performance for two genres: *magazine (mz)* and *pivot (pt)*.

bc

	Ann1	Ann2	Ann3
Ann1	1.0	0.91	0.87
Ann2	0.91	1.0	0.88
Ann3	0.87	0.88	1.0

mz

	Ann1	Ann2	Ann3
Ann1	1.0	0.91	0.94
Ann2	0.91	1.0	0.93
Ann3	0.94	0.93	1.0

pt

	Ann1	Ann2	Ann3
Ann1	1.0	0.97	0.98
Ann2	0.97	1.0	0.96
Ann3	0.98	0.96	1.0

tc

	Ann1	Ann2	Ann3
Ann1	1.0	0.99	0.98
Ann2	0.99	1.0	0.98
Ann3	0.98	0.98	1.0

wb

	Ann1	Ann2	Ann3
Ann1	1.0	0.93	0.90
Ann2	0.93	1.0	0.92
Ann3	0.90	0.92	1.0

Table 11: Genre wise strict decision based disagreement analysis between the annotators.

PoS type	bc	pt
Pronouns	3.6	0.04
Nouns	3.2	0.05
Proper Nouns	1.9	0.03
Verbs	3.5	1.0

Table 12: Genre wise part of speech comparison for two genres: `bc` and `pt`. The numbers are normalized and presented in percentage.

bc

	Ann1	Ann2	Ann3
Ann1	1.0	0.91	0.85
Ann2	0.91	1.0	0.86
Ann3	0.85	0.86	1.0

mz

	Ann1	Ann2	Ann3
Ann1	1.0	0.89	0.91
Ann2	0.89	1.0	0.90
Ann3	0.91	0.90	1.0

pt

	Ann1	Ann2	Ann3
Ann1	1.0	0.94	0.95
Ann2	0.94	1.0	0.91
Ann3	0.95	0.91	1.0

tc

	Ann1	Ann2	Ann3
Ann1	1.0	0.98	0.98
Ann2	0.98	1.0	0.98
Ann3	0.98	0.98	1.0

wb

	Ann1	Ann2	Ann3
Ann1	1.0	0.92	0.90
Ann2	0.92	1.0	0.91
Ann3	0.90	0.91	1.0

Table 13: Genre wise disagreement analysis between the annotators for new chain assignment.

	Memory Size	
Dataset	20	40
OntoNotes	76.6	**77.0**
`LongtoNotes`	72.9	**73.7**
`LongtoNotes`$_s$	70.2	**70.7**

Table 14: **Memory Size vs. Performance**. We compare two settings of the memory size parameter in memory model (Toshniwal et al., 2020b) and find that the larger memory version achieves better results on each dataset.

	OntoNotes						`LongtoNotes`$_s$						`LongtoNotes`					
	Mention			Coref			Mention			Coref			Mention			Coref		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
LongFormer Large (`mz`)																		
+ OntoNotes (384)	88.0	87.9	88.0	82.4	82.4	82.4	84.3	86.1	85.2	73.8	75.0	74.2	84.3	86.1	85.2	73.8	75.0	74.2
+ OntoNotes (4096)	87.9	88.3	88.1	82.4	82.9	**82.6**	84.4	86.7	85.5	74.1	75.9	**74.9**	84.4	86.7	85.5	74.1	75.9	**74.9**
+ `LongtoNotes` (384)	87.0	88.4	87.7	81.4	83.0	**82.2**	84.4	86.9	85.6	72.4	73.6	72.9	84.4	86.9	85.6	72.4	73.6	72.9
+ `LongtoNotes` (4096)	86.9	87.8	87.4	80.9	82.0	81.5	85.0	86.7	85.8	74.1	74.8	**74.4**	85.0	86.7	85.8	74.1	74.8	**74.4**
LongFormer Large (`pt`)																		
+ OntoNotes (384)	95.5	94.4	95.0	88.6	87.4	**88.0**	94.3	95.3	94.8	84.6	86.9	85.7	94.9	94.4	94.7	85.5	85.8	**85.6**
+ OntoNotes (4096)	95.6	94.2	94.9	88.9	86.9	87.9	94.4	94.8	94.6	84.8	86.8	**85.8**	94.9	94.0	94.5	85.5	85.2	85.5
+ `LongtoNotes` (384)	95.1	94.3	94.7	89.2	88.3	88.8	94.2	95.1	94.6	86.0	88.0	**87.0**	94.6	94.2	94.4	86.5	86.7	86.6
+ `LongtoNotes` (4096)	95.3	94.2	94.8	89.7	88.2	**89.0**	94.5	94.5	94.5	86.4	87.4	86.9	94.8	93.7	94.3	87.0	86.4	**86.7**

Table 15: **Comparison of F_1 scores for `mz` and `pt` genres.**

	OntoNotes						`LongtoNotes`$_s$						`LongtoNotes`					
	Mention			Coref			Mention			Coref			Mention			Coref		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
SpanBERT Base (Lee et al., 2017)																		
+ OntoNotes	86.6	87.5	87.0	83.1	83.6	83.4	88.4	85.0	86.7	84.2	80.8	82.4	86.7	85.4	86.1	83.0	81.3	**82.1**
+ `LongtoNotes`$_s$	73.3	91.0	81.2	70.0	85.7	77.1	78.3	90.5	84.0	73.8	85.5	79.2	73.2	90.4	80.9	69.4	85.1	76.5
+ `LongtoNotes`	86.6	87.1	86.8	83.0	82.9	**86.8**	88.1	84.6	86.3	83.3	80.1	81.7	86.6	85.5	86.0	82.4	81.0	81.7
+ `LongtoNotes`$_{eq}$	86.1	87.8	87.0	82.8	83.5	83.2	87.7	86.2	87.0	83.4	81.9	**82.6**	86.1	86.3	86.2	82.3	81.9	**82.1**
LongDocCoref (Toshniwal et al., 2020b)																		
+ OntoNotes	95.3	85.6	86.4	81.2	85.4	83.2	95.3	85.6	86.4	77.8	86.2	81.8	95.3	85.6	86.4	78.2	85.2	81.6
+ `LongtoNotes`$_s$	95.3	85.6	86.4	22.3	66.9	33.5	95.3	85.6	86.4	17.5	65.7	27.6	95.3	85.6	86.4	21.7	66.9	32.8
+ `LongtoNotes`	95.3	85.6	86.4	81.4	85.0	83.2	95.3	85.6	86.4	79.3	85.8	82.4	95.3	85.6	86.4	79.1	85.0	81.9
+ `LongtoNotes`$_{eq}$ (n=3)	95.3	85.6	86.4	81.6	85.2	**83.4**	95.3	85.6	86.4	79.7	86.2	**82.8**	95.3	85.6	86.4	79.3	85.2	**82.2**
+ `LongtoNotes`$_{eq}$ (n=5)	95.3	85.6	86.4	81.4	85.3	83.3	95.3	85.6	86.4	79.7	86.2	**82.8**	95.3	85.6	86.4	79.2	85.3	82.1
+ `LongtoNotes`$_{eq}$ (n=10)	95.3	85.6	86.4	81.5	85.1	83.3	95.3	85.6	86.4	79.7	86.2	**82.8**	95.3	85.6	86.4	79.6	84.8	82.1
+ `LongtoNotes`$_{eq}$ (n=10, m=40)	95.3	85.6	86.4	81.6	85.6	83.6	95.3	85.6	86.4	79.8	85.9	82.7	95.3	85.6	86.4	79.5	85.2	82.3

Table 16: **Comparison of MUC scores**

	OntoNotes						`LongtoNotes`$_s$						`LongtoNotes`					
	Mention			Coref			Mention			Coref			Mention			Coref		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
SpanBERT Base (Lee et al., 2017)																		
+ OntoNotes	86.6	87.5	87.0	75.0	75.5	**75.3**	88.4	85.0	86.7	70.7	65.1	67.8	86.7	85.4	86.1	72.3	69.5	70.9
+ `LongtoNotes`$_s$	73.3	91.0	81.2	57.0	76.8	65.4	78.3	90.5	84	54.8	69.7	61.3	73.2	90.4	80.9	53.3	72.8	61.5
+ `LongtoNotes`	86.6	87.1	86.8	74.6	74.0	74.3	88.1	84.6	86.3	67.5	62.7	65.0	86.6	85.5	86.0	70.6	68.2	69.4
+ `LongtoNotes`$_{eq}$	86.1	87.8	87.0	74.9	75.2	75.0	87.7	86.2	87.0	69.7	67.0	**68.3**	86.1	86.3	86.2	71.7	70.6	**71.2**
LongDocCoref (Toshniwal et al., 2020b)																		
+ OntoNotes	95.3	85.6	86.4	72.2	77.9	74.9	95.3	85.6	86.4	57.9	71.7	64.0	95.3	85.6	86.4	63.9	74.7	68.9
+ `LongtoNotes`$_s$	95.3	85.6	86.4	18.3	61.7	28.2	95.3	85.6	86.4	10.7	53.6	17.9	95.3	85.6	86.4	16.1	58.7	25.2
+ `LongtoNotes`	95.3	85.6	86.4	73.3	76.7	75.0	95.3	85.6	86.4	61.0	70.1	65.2	95.3	85.6	86.4	65.5	73.7	69.4
+`LongtoNotes`$_{eq}$ (n=3)	95.3	85.6	86.4	73.7	76.9	75.2	95.3	85.6	86.4	64.4	70.4	67.3	95.3	85.6	86.4	67.5	73.7	70.5
+ `LongtoNotes`$_{eq}$ (n=5)	95.3	85.6	86.4	73.4	77.3	75.3	95.3	85.6	86.4	64.5	70.9	**67.6**	95.3	85.6	86.4	67.5	74.2	70.7
+ `LongtoNotes`$_{eq}$ (n=10)	95.3	85.6	86.4	73.6	77.0	75.3	95.3	85.6	86.4	64.5	70.9	**67.6**	95.3	85.6	86.4	68.3	73.5	70.8
+ `LongtoNotes`$_{eq}$ (n=10, m=40)	95.3	85.6	86.4	73.5	78.1	**75.7**	95.3	85.6	86.4	65.0	70.5	**67.6**	95.3	85.6	86.4	67.9	74.4	**71.0**

Table 17: **Comparison of BCUB scores**

	OntoNotes						`LongtoNotes`$_s$						`LongtoNotes`					
	Mention			Coref			Mention			Coref			Mention			Coref		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
SpanBERT Base (Lee et al., 2017)																		
+ OntoNotes	86.6	87.5	87.0	71.5	73.7	**72.1**	88.4	85.0	86.7	63.3	61.6	62.4	86.7	85.4	86.1	68.1	68.4	68.2
+ `LongtoNotes`$_s$	73.3	91.0	81.2	53.2	69.5	60.3	78.3	90.5	84.0	51.5	59.2	55.1	73.2	90.4	80.9	50.4	64.2	56.5
+ `LongtoNotes`	86.6	87.1	86.8	70.8	73.1	71.9	88.1	84.6	86.3	63.4	60.5	61.9	86.6	85.5	86.0	67.7	68.2	67.9
+`LongtoNotes`$_{eq}$	86.1	87.8	87.0	70.2	74.2	**72.1**	87.7	86.2	87.0	64.0	63.1	**63.5**	86.1	86.3	86.2	67.5	69.6	**68.5**
LongDocCoref (Toshniwal et al., 2020b)																		
+ OntoNotes	95.3	85.6	86.4	67.0	74.5	70.5	95.3	85.6	86.4	54.5	63.4	58.6	95.3	85.6	86.4	61.6	69.8	65.4
+ `LongtoNotes`$_s$	95.3	85.6	86.4	25.7	60.0	35.9	95.3	85.6	86.4	16.8	47.8	24.8	95.3	85.6	86.4	23.5	57.2	33.3
+ `LongtoNotes`	95.3	85.6	86.4	65.8	75.3	70.2	95.3	85.6	86.4	53.7	65.9	59.2	95.3	85.6	86.4	60.5	71.7	65.6
+ `LongtoNotes`$_{eq}$ (n=3)	95.3	85.6	86.4	66.1	76.2	70.8	95.3	85.6	86.4	54.9	67.4	60.5	95.3	85.6	86.4	61.2	72.2	66.2
+ `LongtoNotes`$_{eq}$ (n=5)	95.3	85.6	86.4	66.7	76.0	71.1	95.3	85.6	86.4	56.0	66.6	60.9	95.3	85.6	86.4	61.9	71.8	66.5
+`LongtoNotes`$_{eq}$ (n=10)	95.3	85.6	86.4	66.2	75.9	70.7	95.3	85.6	86.4	56.0	66.6	60.9	95.3	85.6	86.4	61.7	72.2	66.6
+ `LongtoNotes`$_{eq}$ (n=10, m=40)	95.3	85.6	86.4	68.0	75.9	**71.7**	95.3	85.6	86.4	56.1	68.9	**61.9**	95.3	85.6	86.4	62.9	72.9	**67.5**

Table 18: **Comparison of CEAFE scores**