SPLICE: A Singleton-Enhanced PipeLIne for Coreference REsolution

Yilun Zhu[♠], Siyao Peng[♡], Sameer Pradhan[♣♢], Amir Zeldes[♠]

[♠] Department of Linguistics, Georgetown University
[♡] MaiNLP, Center for Information and Language Processing, LMU Munich
[♣] Linguistic Data Consortium, University of Pennsylvania
[♢] cemantix.org
{yz565, amir.zeldes}@georgetown.edu, siyao.peng@lmu.de, pradhan@cemantix.org

Abstract

Singleton mentions, i.e. entities mentioned only once in a text, are important to how humans understand discourse from a theoretical perspective. However previous attempts to incorporate their detection in end-to-end neural coreference resolution for English have been hampered by the lack of singleton mention spans in the OntoNotes benchmark. This paper addresses this limitation by combining predicted mentions from existing nested NER systems and features derived from OntoNotes syntax trees. With this approach, we create a near approximation of the OntoNotes dataset with all singleton mentions, achieving ∼94% recall on a sample of gold singletons. We then propose a two-step neural mention and coreference resolution system, named SPLICE, and compare its performance to the end-to-end approach in two scenarios: the OntoNotes test set and the out-of-domain (OOD) OntoGUM corpus. Results indicate that reconstructed singleton training yields results comparable to end-to-end systems for OntoNotes, while improving OOD stability (+1.1 avg. F1). We conduct error analysis for mention detection and delve into its impact on coreference clustering, revealing that precision improvements deliver more substantial benefits than increases in recall for resolving coreference chains.

Keywords: Coreference Resolution, Generalization, Mention Detection

1. Introduction

Coreference is a linguistic phenomenon in which two or more expressions (also known as mentions) in a text refer to the same entity (e.g. *the Vice President ... She*). To correctly cluster mentions, the first step is identifying referring expressions, candidates for repeated reference in context. Such mentions include both coreference markables, which are expressions part of a coreference chain, and singletons, which could be referred back to but are not involved in any coreference relations in the given text. From a theoretical linguistic perspective, all mentions are important for coreference resolution because humans understand discourse and entity coherence based on the competing available options (Grosz et al., 1995). From an empirical perspective, both markables and singletons are important components in the data distribution for cluster linking, with coreference markables corresponding to true positives and singletons corresponding to true negatives (Kübler and Zhekova, 2011), while all mentions can be used to improve coreference markable boundary detection.

In recent years, end-to-end (Lee et al., 2017, 2018) and sequence-to-sequence (Bohnet et al., 2023) approaches have demonstrated superior performance compared to rule-based (Raghunathan et al., 2010; Lee et al., 2013) and entity-based neural approaches (Wiseman et al., 2015; Clark and Manning, 2015, 2016a,b). Despite progress achieved by deep learning models, the proposed solutions diverge from discourse theory by not considering all mention candidates, particularly singletons, when learning coreference linking, which also results in limited interpretability of existing models. The major reason that this issue has been overlooked is that the dataset used for training most coreference resolution systems, OntoNotes (Pradhan et al., 2013), lacks singleton annotations.[1]

Previous work on recovering singletons in the data initially used gold syntax annotations from OntoNotes to develop a rule-based algorithm. Raghunathan et al. (2010) extracted pronouns and maximal NP projections and incorporated a post-processing step that employed a set of rules to filter out mentions that didn't align with the annotations, such as numeric mentions. Such approaches have been used as a preprocessing step for several coreference systems (Wiseman et al., 2015, 2016). Another strategy involves parsing syntax trees to identify all NPs from the corpus

[1] There were two main reasons for not annotating singletons in OntoNotes: i) Annotating singletons would have increased the annotation effort significantly; therefore, a trade-off had to be made; ii) Inter-annotator agreement on whether or not a text span is referential was also relatively low.

LREC-COLING 2024, pages 15191–15201
20-25 May, 2024.

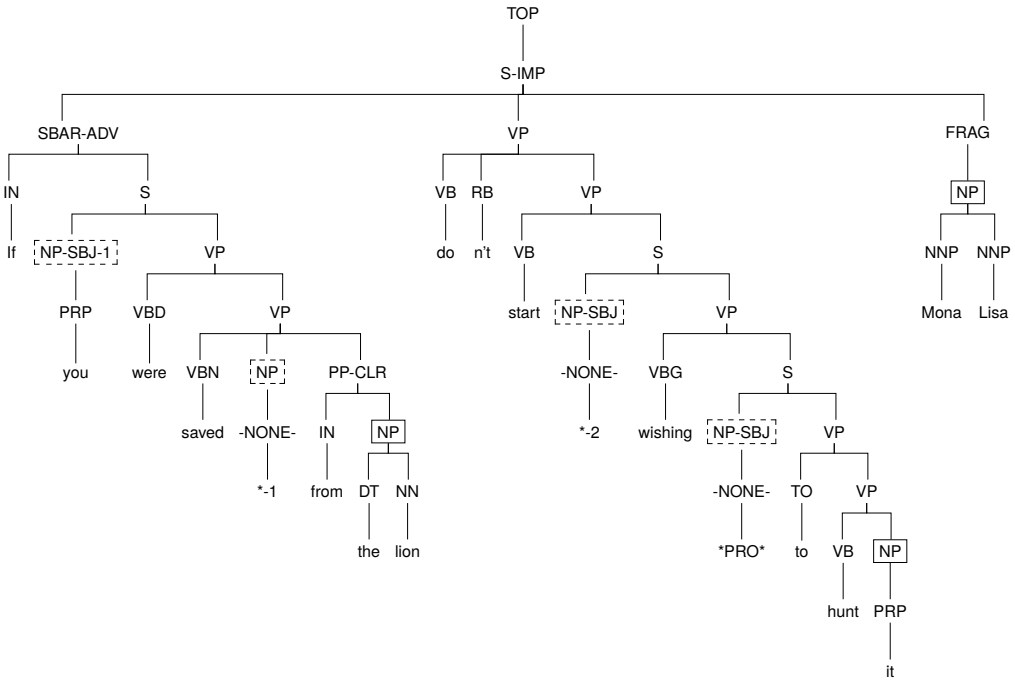

Figure 1: An example of the utilization of a syntax tree for the extraction of mentions. The solid box signifies that the NP is a candidate for coreference linking in OntoNotes while the dashed box indicates that the NP is not categorized as a mention.

(Clark and Manning, 2015, 2016b), which was frequently used before the advent of end-to-end systems. The third method aims to generate silver singletons for the corpus. Recasens et al. (2013) proposed a lifespan model to make distinctions between singleton mentions and coreference markables. More recently, Toshniwal et al. (2021) proposed a data augmentation strategy to extract silver mentions ('pseudo-singletons') from a mention detector trained on OntoNotes coreference markables. Zhu et al. (2023) demonstrated that gold singletons and mention-based features can improve coreference scores on OntoGUM, an out-of-domain (OOD) dataset following the OntoNotes scheme.

However, these methods exhibit certain limitations. While extracting NP subtrees can achieve a high recall in singleton detection, it concurrently generates a large number of precision errors (spans that are not valid mentions). For example, the span *you* in Figure 1 is a valid NP but is not a mention candidate for pair matching as it is a *generic* you. By contrast, the second method is trained to pick up OntoNotes mentions, but falls short due to two reasons: (1) the system is biased towards mentions that resemble coreference markables in OntoNotes, missing atypical ones with semantic and syntactic disparities; (2) evaluating performance of the mention detector is challenging without any gold singletons, meaning we do not know how its performance is impacting downstream coreference scores. Finally, we note that in realistic settings, applications may want access to all entities mentioned in a text, including singletons, meaning their comprehensive detection is desirable.

Based on the necessity of singletons for context understanding and the existing problems of previous research, our paper aims to extract near-gold singletons using datasets with singleton annotations. We demonstrate how to effectively employ these in coreference systems to improve in/out-of-domain performance. The contributions[2] include:

- A mention detection classifier that extracts mentions from syntactic structures and achieves a recall of ∼94% on the OntoNotes development set.
- A near-gold singleton annotated version of OntoNotes.
- A pipeline-based neural coreference system, named SPLICE, using singletons, yielding results on par with the end-to-end approach in-domain and a +1.1 boost OOD.
- Evaluation at different precision and recall levels for mention detection and analysis of the effect of singletons on coreference linking.

[2]The code for training the mention classifier & the coreference pipeline and data of OntoNotes singletons are publicly available at https://github.com/yilunzhu/splice.

2. Related Work

Mention Detection As a preprocessing step, mention detection is an important component of coreference resolution. Most neural approaches implement mention detection as part of an end-to-end system. The widely-used pre-neural OntoNotes mention detector employed a rule-based system to extract pronouns and maximal NP projections given gold syntax trees (Raghunathan et al., 2010), while the end-to-end approach from Lee et al. (2017, 2018) detected markables directly during pair matching. Yu et al. (2020) compared three neural approaches for scoring markable spans and proposed using a biaffine model, which achieved a high recall on mention detection.

Coreference Resolution Recent neural coreference resolution systems have achieved great improvements. The end-to-end approach (Lee et al., 2017) jointly learns mention detection and coreference pair scoring and achieved SOTA scores on the OntoNotes test set before several extensions were proposed. Lee et al. (2018); Kantor and Globerson (2019) improved span representations to improve pair matching. Joshi et al. (2020) added better pre-trained language models to gain additional score boosts. Wu et al. (2020) adapted a question-answering framework into the task and improved both coreference markable span detection and coreference matching scores. Dobrovolskii (2021) also improved performance by initially matching coreference links via words instead of spans. Recently, Bohnet et al. (2023) proposed a sequence-to-sequence paradigm to predict mentions and links jointly. However, none of these models consider singletons for coreference linking.

3. Nominal Phrase Extraction

Mentions are typically manifested as noun phrases (NPs) in syntactic structures.[3] However, due to the intricate nature of NPs, whose recognition entails prepositional phrase (PP) attachment disambiguation and diverse sentence structure analyses, a considerable portion of NPs is excluded from consideration as valid mentions for subsequent coreference resolution according to the annotation guidelines of OntoNotes,[4] such as nested mentions inside proper names, generic *you*, expletives, adjectives and non-proper nouns within pre-modifiers, etc. For example, the sentence in Figure 1 contains 7 NPs. Three of them are syntactic traces, marked by *-NONE-* in the node, and one is a pronominal phrase without phonological content, marked as **PRO**, both of which are discarded during the pre-processing step. There are three possible coreferential mentions: *the lion*, *it*, and *Mona Lisa*. By contrast, the NP *you*, cannot be considered a potential mention during linking since generic *you* is not annotated as a coreference markable in OntoNotes. To prevent the inclusion of "invalid" mentions during the coreference linking stage, excluding these NPs from the list of potential mention candidates is helpful.

3.1. Dataset Preparation for NPs

In the coreference layer, OntoNotes does not have singleton annotation. Although the corpus also has named entity annotations, these cannot solve the problem because only flat named entities are annotated (no non-named or nested mentions). However, not only flat named entities, but also nested, named, and non-named entities (=coreference markables + singletons) are candidates for coreference linking according to OntoNotes annotation schema. When constructing the corpus, annotators were offered gold pronouns and NPs as coreference markable candidates, meaning gold syntax trees could be utilized to recall near-gold singletons. The precision of mapping syntactic NPs to singletons could be studied using data with both annotation types. To the best of our knowledge, only ARRAU (Poesio et al., 2018) and OntoGUM (Zhu et al., 2021) meet these requirements.[5] Among the four genres in the ARRAU corpus, the RST news genre consists of the subset of the Penn Treebank (PTB, Marcus et al. 1993) that was annotated for discourse relations in the RST treebank (Carlson et al., 2001). Since OntoNotes overlaps the same subset of PTB, ARRAU can be used to compare singleton annotations with PTB constituent trees, though its coreference scheme differs from OntoNotes. OntoGUM, designed to follow the OntoNotes annotation scheme and featuring 12 genres, is a second option, though it has predicted constituent trees converted from gold dependency structures. Thus, OntoGUM can provide OOD data to make the classifier more robust across genres/datasets. OntoGUM V9 is utilized in this paper. We use the two datasets to create a classification model to map gold NPs to near-gold singletons. Since RST documents overlap OntoNotes, we rearrange document splits to facilitate downstream mention classification and coreference: train/test: 265/148.[6]

[3] Some coreference guidelines also consider verbs referred back to by NPs (which are then a fraction of all mentions) as mention candidates.

[4] https://catalog.ldc.upenn.edu/LDC2013T19

[5] ARRAU annotates more mentions compared to OntoNotes, including non-referring expressions, such as *on [the other hand]*.

[6] We release our re-split together with our model.

Category	P	R	F1	Num.
0	0.92	0.84	0.87	8,087
1	0.95	**0.97**	0.96	23,877
Micro Avg	0.94	0.94	0.94	31,964

Table 1: Results of the XGBoost NP classifier on the test set (new data split excluding OntoNotes test documents) of ARRAU. **1** denotes the NP is a mention and **0** presents the opposite.

3.2. Mention Classification

Let $I = \{1, ..., i\}$ be the number of NPs and pronouns within a document. This classification task aims to distinguish mentions that can potentially have coreference relationships (referring expressions) from other NPs.

We use ARRAU's RST portion with the above split for training and evaluation. First, all pronouns and NPs are extracted from the gold syntax trees so that each span can be allocated a label based on ARRAU's coreference layer annotations. Pronouns identified through a rule-based function and NPs annotated within ARRAU are assigned positive labels. Other NPs are assigned negative labels.

For effective training of an NP classifier, relying solely on information within mentions spans is insufficient: NP that would be mentions in one context may not be when nested in another phrase. Furthermore, the variable count of parent and child spans in each phrase introduces challenges if we want to avoid handpicked features. To address these problems, we introduce a set of generic features that describe each NP span and its graph position. The features for each NP consist of two primary components. The first set of features encompasses mention-based features of the current NP, its parent phrases, and child phrases. These features include parts-of-speech tags, the usage of prepositions, definite markers, grammatical roles, adverbial tags, etc. Additionally, we extract features from other NPs that overlap with the current one, considering features like their relative positions or hierarchical levels among other NPs, as well as the largest and smallest interactive NP spans. We select the XGBoost classifier (Chen and Guestrin, 2016) for the NP classification task and evaluate its performance on the ARRAU test set. We train the `gbtree` booster with a `learning_rate` set at 0.1. Due to some disparities between OntoNotes markables and PTB NPs, a small portion of mentions cannot be extracted via syntax structures, for example, compound modifiers such as '[Hong Kong] government.' In such cases, we introduce a post-processing step to recover non-NP singletons.

Table 1 shows that the classifier performs well on ARRAU, demonstrating the model's capability to map PTB tree patterns to distinguish potential coreference markables. In particular, the model excels in recall of coreference markables. A recall-focused model offers the advantage of generating a substantial pool of candidates for potential linking, affording the coreference resolver a secondary opportunity to eliminate non-mention spans.

Dataset	P	R	F1
ARRAU	28.15	97.78	44.35
OntoNotes	39.46	91.65	55.16

Table 2: Results of coreference markables on ARRAU and OntoNotes test captured by the NP classifier.

Additionally, we want to evaluate the classifier's ability to identify coreference markables. Table 2 presents the scores for coreference markables as captured by the classifier both in-domain (ARRAU) and OOD (OntoNotes). Given the minor disparities in annotation and genre between the two datasets, the classifier performs well (here we focus on the recall score) in distinguishing mentions from NPs. The predictions rendered by the classifier are utilized in training the mention detector and is available at the public GitHub link introduced in footnote 2.

4. Coreference Pipeline

With the enhanced OntoNotes data in hand, we build a training pipeline for coreference inference.

4.1. Dataset

OntoNotes V5.0 (Pradhan et al., 2013) contains 1.6M tokens annotated for coreference, with a test set comprising 348 documents with ∼170K tokens. We also use OntoGUM V8.0 (Zhu et al., 2021) as an OOD dataset to evaluate the model's generalization performance. OntoGUM's test set includes 24 documents with ∼22K tokens, following the same coreference annotation scheme as OntoNotes.

4.2. Training

Mention detector We use the classifier trained on ARRAU to predict positive and negative labels within the OntoNotes training dataset. Then, we take the union of the classifier's outputs and gold coreference markables from the OntoNotes train set. This data serves as input for the mention detector. We realize that the mention span detection task closely resembles the nested named-entity recognition (NNER) task, with the key distinction being that mention detection does not require entity types. Thus, we train a SOTA NNER system (Tan et al., 2021) on the union data, which we will use to predict mentions at test time, as shown in Figure 2.

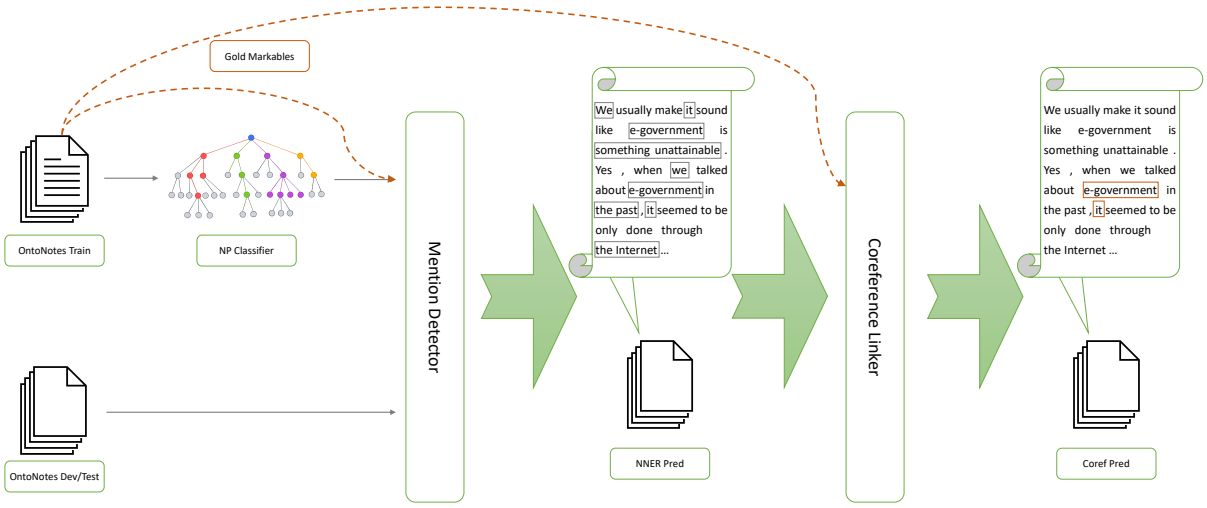

Figure 2: The Pipeline of the Two-step Coreference System Using Singletons. Gold markable spans are leveraged for training mention detection and coreference linking to enhance alignment with the OntoNotes annotation schema.

Data	Precision		Recall		F1
OntoNotes-dev	37.84	(18,321/48,419)	95.64	(18,321/19,156)	54.22
OntoGUM-test	37.75	(19,018/50,736)	96.23	(19,018/19,764)	54.23
OntoGUM-test	37.21	(2,439/ 6,554)	91.66	(2,439/ 2,661)	52.94

Table 3: Mention detection performance on OntoNotes dev/test set and OntoGUM test set.

The performance results are presented in Table 3. The mention detector demonstrates a high recall rate, identifying approximately 96% of coreference markables in both the validation and test sets of OntoNotes. It reveals that it effectively captures a significant portion of the relevant information according to OntoNotes' guidelines. However, the precision score is comparatively lower at ∼37.8%. Since there are approximately twice as many mentions as coreference markables (Zhu et al., 2021), this suggests an estimated count of around 40K gold mentions in OntoNotes dev and test sets each. However, the currently extracted mentions contain nearly 20% ($10\text{K} = 48\text{K} - 19\text{K} * 2$) of incorrectly predicted spans. Though the mention classifier and the mention detector achieve the best results in extracting near-gold singleton spans from OntoNotes, the low precision score for coreference markables indicates that a significant number of identified mentions may distract the coreference linker.

The extracted mentions are then used in the coreference model training process, which learns to identify and link valid mentions implicitly. In OOD evaluation, it is also observed that the mention detector produces high-quality mentions with a recall of nearly 92%.

Coreference model We use the end-to-end (e2e) model with SpanBERT-large embeddings (Joshi et al., 2020) as our baseline model. The baseline considers all span possibilities during coreference linking. As in (1), it uses a feed-forward network to compute a markable score for each possible token span, represented by a concatenation of four vectors: token embedding of the start token and the end token of the span, attention-based head embeddings, and meta information (such as genres, gold speaker information) shown in (2). The neural network calculates a mention score for each span. It then keeps a fixed number of spans with top scores for coreference clustering. However, this pruning method may exclude correct markable spans and singletons with lower scores during inference. For this reason, our proposed pipeline diverges by exclusively training on the union of gold coreference markables and positive outputs from the mention detector. We then assign identical mention scores to all spans, ensuring that the likelihood of span mentions does not influence the model's coreference linking decisions. As illustrated in (3), we utilize a trainable parameter w_m for the markable score and employ hyperparameter tuning to find the best alignment with coreference clustering.

(1)

Baseline: $s_m = \text{FFNN}_\text{m}(g_i)$

(2)

$$g_i = [x_{\texttt{start(i)}}, x_{\texttt{end(i)}}, \hat{x}_{\texttt{i}}, \phi(i)]$$

(3)

$$\text{Ours: } s_m = w_m$$

To maintain consistency in training experiments, we keep other hyperparameters at the same values as the baseline model, mitigating the impact of external factors. The proposed pipeline for training and inference is outlined in Figure 2.

4.3. Inference

The evaluation of the coreference resolution task requires plain text as input, meaning mention spans and gold syntax trees cannot be used at test time. Consequently, as shown in Figure 2, inference is divided into two steps: First, the mention detector (based on an NNER system) reads the plain input and generates nested mentions. Second, these predicted mention spans are provided as input to the coreference model, which constructs coreference chains based on them.

5. Experiments

5.1. Experiments Setup

Following Tan et al. (2021), the mention detector uses BERT-base (Devlin et al., 2019) as the base model to train the system. All coreference clustering experiments use Pytorch and the pre-trained SpanBERT large (Joshi et al., 2020) model from HuggingFace[7] for token representations. Both mention detection and coreference linking experiments are conducted on Nvidia RTX A6000 GPUs with 64GB RAM.

5.2. Results

In-domain Table 4 presents the results of our proposed pipeline and the baseline model on the OntoNotes test set. When using the predictions from the mention detector (`Ours+MD`), our model yields comparable mention spans and achieves a comparable average F1 score to the baseline model (79.4 vs. 79.6). This indicates our model can effectively learn to resolve coreference even when provided with imperfect input. Additionally, the results suggest that with recent improvements in nested NER systems, a sufficient number of coreference markables can be captured, making the end-to-end architecture not significantly superior to our pipeline-based system. More than that, our pipeline-based model can output all entities: singletons and coreferring chains.

Using both gold coreference markables and predicted mentions (`Ours+MD+GM`) as inputs represents an upper bound for our pipeline-based system. Our model achieves an average F1 score of 83.5 in this scenario, marking a nearly 4-point increase over the baseline. This substantial gap indicates that, although the mention detection module generates some incorrect spans (precision errors), the coreference clustering module can generally construct correct clusters from the provided spans.

These results demonstrate that our system is a robust approach to coreference resolution in-domain. We further assess how stable OOD is below.

OOD performance Table 5 compares the baseline model and our system for OOD evaluation on the OntoGUM test set, which includes 12 written and spoken genres, especially challenging ones such as conversation, fiction, and YouTube vlog transcripts. The results indicate that the predictions of the mention detector (`Ours+MD`) provide improved mention detection scores, showing an increase of 1.4 points compared to the baseline model. Consequently, the model achieves an average F1 score of 66.4, outperforming the baseline model by 1.1 points. The improvement demonstrates the effectiveness of our proposed pipeline, which enhances the generalization on unseen data.

Because OntoGUM contains gold singleton annotation, we can directly use this mention information to assess the performance of the coreference clustering module within e2e systems. With gold singletons (`Ours+GS`), our pipeline achieves an average F1 score of 70.8, resulting in a larger improvement of 5.5 points over the baseline model (65.3). This improvement gap highlights the importance of gold singleton annotations.

5.3. Analysis

5.3.1. Qualitative Analysis

We conduct a qualitative analysis comparing our mention predictions to the gold spans in OntoNotes.

Recall The mention detector misses 4.36% of coreference markables on ON-dev as in Table 3. We manually categorize these into five groups, as exemplified in Table 6. The first type of error relates to `missing nested entities`. For example, the nested and coordinated "bridge-"modifier, "Zhuhai - Hong Kong - Macao," and the second city "Hong Kong" are missing from predictions. The second type `attachment of prepositional phrases` also relates to complex NPs. Particularly, when a noun interacts with PP attachments,

[7] https://huggingface.co/

	Mention Detection			MUC			B^3			$CEAF_{\phi4}$			Avg. F1
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
Joshi et al. (2020)	89.1	86.5	87.8	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	79.6
Ours+MD	88.8	87.3	88.1	85.6	84.5	85.1	78.8	77.0	77.9	75.8	74.4	75.1	79.4
Ours+MD+GM (upperbound)	90.9	91.3	91.1	87.9	88.6	88.3	81.4	82.7	82.0	80.3	79.9	80.1	83.5

Table 4: Results on OntoNotes test set. *MD* denotes the model uses predictions from the mention detector; *GM* indicates the model uses gold coreference markables.

	Mention Detection			MUC			B^3			$CEAF_{\phi4}$			Avg. F1
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
Joshi et al. (2020)	86.0	70.6	77.5	80.0	68.1	73.6	67.9	60.5	64.0	68.6	50.5	58.2	65.3
Ours+MD	85.3	73.5	78.9	78.8	70.6	74.5	66.5	63.5	64.9	68.3	52.0	59.0	66.4
Ours+GS (upperbound)	90.8	74.8	82.0	84.8	72.4	78.1	74.2	65.6	69.6	75.7	55.6	64.2	70.8

Table 5: Results on OntoGUM test set. *GS* indicates that our model uses gold singletons.

the mention detector occasionally fails to make accurate predictions. In addition, NPs are complex in garden-path sentences, challenging both the mention detector and human comprehension. Fourth, while verbs can have coreference relations in OntoNotes, they comprise a very small portion (less than 2%) of the annotated data. Consequently, due to underrepresentation, the mention detector, which knows these only from the unioned gold coreference data, may miss some verbal markables. The remaining type is gold annotation errors. Coreference in OntoNotes relies heavily on syntax trees, particularly NP spans, and annotations in OntoNotes are not always correct. Some entities are incorrectly split into two parts due to annotation errors in the syntax tree. These splits are typically observed in post-nominal prepositional phrases and relative clauses. Additionally, redundant punctuations, such as extra commas and opening quotation marks, are sometimes incorrectly included.

The manual inspection reveals that recall errors are distributed sporadically, and a considerable subset of them cannot be avoided even with a better mention detector. Consequently, addressing these remains a substantial challenge and requires significant effort to bridge the gap.

Precision Due to the absence of singletons in OntoNotes annotations, we estimate that only ∼20% of the missing precision is relevant to mention detection errors (see Sec 4.2). Three types of precision errors were observed, as shown in Table 6. These errors include redundant punctuations as in "one ." or redundant non-restrictive relative clauses as in "5 p.m. EST – when stocks there plunged." The most tricky cases are generic NPs that do not refer to specific mentions. These include negative phrases such as "no media" or quantifier phrases such as "any of the Disney symbols." Though manually spotting such precision errors provides some insight, the lack of singleton annotations in OntoNotes hinders the possibility of exhaustively quantifying precision error types. This underscores the value of gold-singleton annotated corpora for mention detection evaluation.

Recall
Missing nested entity: Once the [Zhuhai - [Hong Kong] - Macao] bridge is built ...
Attachment of prepositional phrases: He just told [a story] uh from the beginning to the end.
Garden-path sentences: Like [the bones] xrays of his wisdom teeth also tell us something about his age.
Missing verbal referents: ... a unit of Marines [killed]$_{\#126}$ some 24 unarmed Iraqis ... [this atrocity]$_{\#126}$ ...
Gold annotation errors: They can volunteer at [any] [of thousands of non-profit institutions] ...
Precision
Redundant punctuations: [one .]
Redundant non-restrictive relative clauses: [5 p.m. EST – when stocks there plunged.]
Generic NPs: [no media]

Table 6: Major categories of recall and precision errors in OntoNotes dev set. [Square brackets] denotes gold mention spans and underlining indicates the most relevant predicted span (if necessary). Precision errors are enclosed by [square brackets].

5.3.2. Effect of Mention Detection

One of the advantages of the proposed pipeline is that the two separate steps provide more transparency than e2e models. Consequently, we investigate how mention detection affects coreference linking.

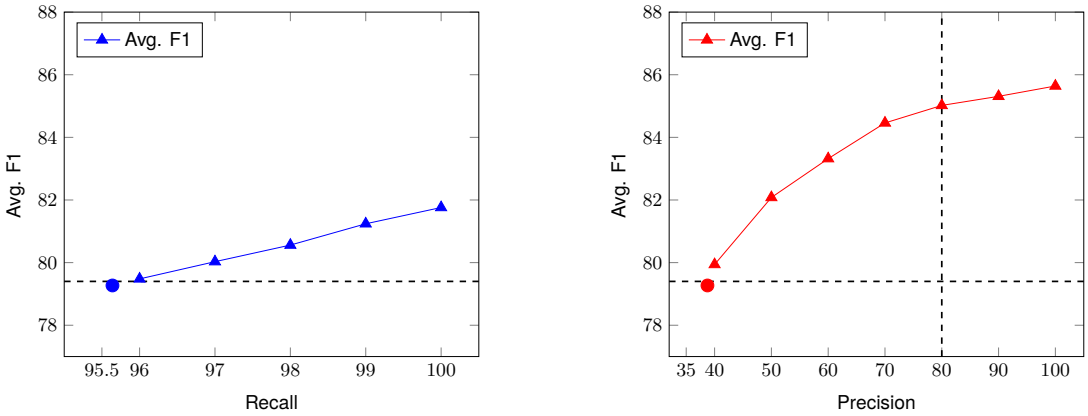

Figure 3: Analyzing the impact of recall and precision scores on the OntoNotes development set. The horizontal dashed line represents the baseline score and the rounded data point denotes the F1 scores achieved by the two-step training pipeline, aligned with their respective precision and recall scores. The vertical dashed line denotes an estimation of avg. F1 and precision score with gold singletons.

Recall The left side of Figure 3 demonstrates the impact of recall on coreference resolution in the OntoNotes development set. We focus on the relationship between mention recall and coreference resolution performance.

Compared to precision, increasing the coverage of gold markables is a more intuitive way to benefit coreference clustering: markables that are undetected will inevitably lead to errors. Therefore, to investigate the significance of recall in coreference clustering, we explore different recall scenarios by adjusting the proportion of recall errors in the mention spans provided to the coreference module. We randomly omit certain portions of the recall errors to improve the recall score. These spans are evenly distributed among the documents in the OntoNotes development set. Initially, the mention detector's recall score stands at around 95.6%, meaning it correctly identifies the majority of coreference markables. We then gradually increase the recall score from 96 to 100. $Recall = 100$ indicates that all gold coreference markables are covered by predicted mentions.

The left panel in Figure 3 also shows the average F1 score for various recall scores (number of mentions) within the OntoNotes development set. It exhibits a similar trend to the precision plot: with an increase in recall scores, the average F1 score also increases. When all recall errors are removed, the coreference score achieves an average F1 of 81.8, increasing the initial score by 2.4 points and outperforming the baseline model by 2.2 points, a substantial improvement but not as large as the maximal precision-based improvement.

Precision The results from Table 3 demonstrate that the mention detector produces a substantial amount of precision errors, indicating that it identifies many potential mentions that are not a coreference markable. Such precision errors are unsurprising since singletons are excluded from coreference markables. Still, they can lead to the coreference resolution model making incorrect predictions, even if the model performs well in other respects.

To thoroughly explore how mention detection influences coreference clustering, like the recall plot, we modify the number of incorrect mentions from the mention detector's predictions and feed various mention spans to the coreference module, using the gold annotations to distinguish markables that will corefer with others (as shown in the right side of Figure 3). Specifically, we randomly eliminate incorrect markables from the OntoNotes development set to enhance the precision score. For instance, the initial precision score of the mention detector is 37.8 (as represented by the red circular data point), indicating that it retrieves around 48K mentions, of which 18K are accurate coreference markables. To increase the precision score to 50%, we select 2.6K erroneous spans and distribute them equally among development documents. Starting from 37.8, the adjusted precision score ranges from 40 to 100. It is noted that $precision = 100$ represents an ideal case, indicating that all predicted mentions are gold coreference markables.

The right panel of Figure 3 represents the average F1 score as a function of the precision score (number of mentions) within the OntoNotes development set. Unsurprisingly, higher mention precision scores correlate with enhanced average F1 scores. When we selectively remove 11% of the precision errors ($\sim$5.6K) from the predicted mentions list by increasing the precision score from 39 (the circle point) to 50, we observe an average F1 score of 2.7 points increase, indicating a substantial improvement over the baseline model.

As discussed in Section 4.2, we estimate around 40K mentions (including singletons) in the

OntoNotes development set. Consequently, we can approach an upper bound on the precision score in our context. Given the number of predicted mentions (48K), we can, at most, randomly remove around 8K spans, resulting in an 80% precision score (illustrated by the vertical dashed line in the figure). Given the current mention detector with nearly 96% recall, the model can achieve the best coreference score, reaching an F1 of 85.0 with a precision of 80%. This analysis demonstrates the critical role of mention precision in coreference resolution and its potential to impact model performance.

In sum, reducing both mention precision and recall errors substantially impacts coreference resolution performance. While the coreference clustering module can handle some incorrect spans by rejecting them during linking, certain precision errors continue to affect its performance. Furthermore, since existing systems already achieve high recall scores for mention detection, increasing recall further is challenging. By contrast, the precision score is still relatively low and can be improved more easily. Thus, focusing on precision improvement will likely offer more significant benefits for future coreference models.

6. Conclusion

This paper introduces a novel approach to address the coreference resolution challenge. It establishes a near-gold singleton dataset for OntoNotes, which is shown to be highly accurate. This dataset can benefit further research endeavors involving singletons in coreference systems. Additionally, we propose SPLICE, a pipeline-based neural system that independently trains mention detection and coreference models. Our system achieves comparable in-domain results with the e2e approach and demonstrates superior OOD performance. Leveraging the better interpretability of our system, we conduct a comprehensive analysis of mention predictions. We discover that resolving additional recall errors is more challenging than addressing precision errors, which offers valuable insight for future work in coreference resolution research.

7. Limitations

This work does not provide a manual validation of the performance of the singletons constructed by the NP classifier and the mention detector in OntoNotes, though we do provide an evaluation on the gold standard annotations in ARRAU.

The focus of this paper is on the NP to mention classification task for English datasets, and it might not cover certain linguistic phenomena found more often in other languages, such as zero anaphora in Chinese and Spanish. Furthermore, the performance of NP classification or mention detection aligning with data in other languages included in OntoNotes, such as Arabic (Pradhan et al., 2013) or Chinese (Pradhan et al., 2013), and multiple languages in multilingual coreference benchmarks such as CorefUD (Nedoluzhko et al., 2022), has not been evaluated in this work.

This paper evaluates the system's generalizability on the OntoGUM corpus. There are other challenging coreference datasets, such as GENTLE (Aoyama et al., 2023), not included in our OOD evaluation.

8. Bibliographical References

Tatsuya Aoyama, Shabnam Behzad, Luke Gessler, Lauren Levine, Jessica Lin, Yang Janet Liu, Siyao Peng, Yilun Zhu, and Amir Zeldes. 2023. GENTLE: A genre-diverse multilayer challenge set for English NLP and linguistic evaluation. In *Proceedings of the 17th Linguistic Annotation Workshop (LAW-XVII)*, pages 166–178, Toronto, Canada. Association for Computational Linguistics.

Bernd Bohnet, Chris Alberti, and Michael Collins. 2023. Coreference resolution through a seq2seq transition-based system. *Transactions of the Association for Computational Linguistics*, 11:212–226.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1405–1415, Beijing, China. 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.

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

Vladimir Dobrovolskii. 2021. Word-level coreference resolution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7670–7675, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

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.

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

Sandra Kübler and Desislava Zhekova. 2011. Singletons and coreference resolution evaluation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 261–267, Hissar, Bulgaria. Association for Computational Linguistics.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. 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. 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.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Anna Nedoluzhko, Michal Novák, Martin Popel, Zdeněk Žabokrtský, Amir Zeldes, and Daniel Zeman. 2022. CorefUD 1.0: Coreference meets Universal Dependencies. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4859–4872, Marseille, France. European Language Resources Association.

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.

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.

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, Marie-Catherine de Marneffe, and Christopher Potts. 2013. The life and death of discourse entities: Identifying singleton mentions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 627–633, Atlanta, Georgia. Association for Computational Linguistics.

Zeqi Tan, Yongliang Shen, Shuai Zhang, Weiming Lu, and Yueting Zhuang. 2021. A sequence-to-set network for nested named entity recognition.

Shubham Toshniwal, Patrick Xia, Sam Wiseman, Karen Livescu, and Kevin Gimpel. 2021. On generalization in coreference resolution. In *Proceedings of the Fourth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 111–120, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sam Wiseman, Alexander M. Rush, Stuart Shieber, and Jason Weston. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1416–1426, Beijing, China. Association for Computational Linguistics.

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.

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.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Neural mention detection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1–10, Marseille, France. European Language Resources Association.

Yilun Zhu, Siyao Peng, Sameer Pradhan, and Amir Zeldes. 2023. Incorporating singletons and mention-based features in coreference resolution via multi-task learning for better generalization. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 121–130, Nusa Dua, Bali. Association for Computational Linguistics.

Yilun Zhu, Sameer Pradhan, and Amir Zeldes. 2021. OntoGUM: Evaluating contextualized SOTA coreference resolution on 12 more genres. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 461–467, Online. Association for Computational Linguistics.