COREFERENCE RESOLUTION FOR DOWNSTREAM NLP TASKS

By

Sushanta Kumar Pani

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science – Master of Science

2021

ABSTRACT

COREFERENCE RESOLUTION FOR DOWNSTREAM NLP TASKS

By

Sushanta Kumar Pani

Natural Language Processing (NLP) tasks have witnessed a significant improvement in performance by utilizing the power of end-to-end neural network models. An NLP system built for one job can contribute to other closely related tasks. Coreference Resolution (CR) systems work on resolving references and are at the core of many NLP tasks. The coreference resolution refers to the linking of repeated object references in a text. CR systems can boost the performance of downstream NLP tasks, such as Text Summarization, Question Answering, Machine Translation, etc. We provide a detailed comparative error analysis of two state-of-the-art coreference resolution systems to understand error distribution in the predicted output. The understanding of error distribution is helpful to interpret the system behavior. Eventually, this will contribute to the selection of an optimal CR system for a specific target task.

Dedicated to Maa and Bapa

ACKNOWLEDGMENTS

First of all, I would like to offer sincere gratitude to my advisor Dr. Parisa Kordjamshidi for her guidance, support, and immense patience. It has been my privilege to have Dr. Jiliang Tang, Dr. Kristen Johnson, and Dr. Hamid Karimian being on my committee, and thankful for their guidance. I thank Professor Joyce Chai for guiding me during my initial days at MSU. I thank Professor Arun Ross, Dr. Karthik Durvasula, Professor Pann-Ning Tan, Professor Xiaoming Liu, and Professor Li Xiao for their courses that helped me improve my skills. I am thankful to Dr. Eric Torng and Professor Sandeep Kulkarni for their help and encouragement. I thank Dr. Katy Colbry for her research writing guidance. I want to thank Steven R. Smith, Amy King, Brenda Hodge, and Erin Dunlop for their support and help.

I am blessed to have an awesome bunch of friends as LIAR and HLR group lab mates in the order of acquaintance; Shane Storks, Sari Saba-Sadiya, Dr. Qiaozi Gao, Dr. Shaohua Yang, Guangyue Xu, James Peterkin II, Dr. Quan Guo, Hossein Faghihi, Drew Hayward, Chen Zheng, Roshanak Mirzaee, Yue Zhang, Dr. Elaheh Raisi, Darius Nafar, and Tim Moran. I thank friends from Michigan State University: Sudarshan, Affan, Apoorva, Gauri, Sneha, Aditya, and Shalin for making life lively and colorful. I offer my sincere obeisances to ISKCON Lansing Family for their guidance and support throughout my graduate study period. I appreciate the contribution of colleagues and friends in India towards my professional as well as personal progress. I take this opportunity to pay the highest obeisances to all my teachers, guides, and well-wishers. I am highly indebted to the love and care of Sri. Bhismadev Nag and Sri Purusottam Pati.

I thank everyone from both of my families for always being supportive. In particular, I thank Maa and Bapa, who are always there for me. Life is always full of pleasure with my siblings: Nani, Ruby, and Bhai. I thank Bhaina, Dada, and Bhauja for keeping my morals high always. I thank the nextgen in our family: Om, Subh, Sonu, Mona, Sona, and Subham for their love and affection. I thank God for the gift as a new family with Mummy, Daddy, Poonam Di, Jop ji, Vishnu, Chetna, and of course Rahul, Anika, and Jia+.... Life is awesome with all of you.

I admire my friend and wife Renu for being the reservoir of encouragement contributing to my life progress.

I love everything due to Krishna...

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Natural Language Processing (NLP) has witnessed performance advancement in many tasks, such as Information Extraction, Question Answering, and Text Summarization. However, most of these systems have faced challenges in resolving references. The ambiguity of reference in this system can be minimized by using a Coreference Resolution (CR) to achieve a higher level of accuracy [34]. Several methods have been utilized for coreference resolution. CR is difficult, which is evident from the performance of these methods on commonly used evaluation metrics. There is a good scope of work in the rectification of dataset annotation (missing and incomplete) and improvement of evaluation methodology used in CR. Recent end-to-end CR systems [23, 24, 28, 29] designed using the power of deep learning algorithms have a primary focus on improving accuracy. However, a detailed error analysis is required [25] to compare these models. This comparative analysis is helpful to select a system for a specific downstream task. The error analysis is also contributing to the interpretation of the CR system's decisions.

In this thesis, we work on two state-of-the-art end-to-end CR systems: BERT-base and SpanBERT-base [23, 24]. We compare these system performances on CoNLL 2012 shared task data [36]. We also perform detailed error analysis and compute corresponding error distributions of these models motivated by the work of Kummerfeld and Klein [25].

1.1 Coreference Resolution

Language-based communication has a significant contribution to the progress of the human race. Speech and text are two core elements of language-based human communication. A good level of understanding is required about entities from their reference (mentions) in speech or text. Humans easily communicate due to their ability to comprehend entity references. This is still an arduous challenge for computing-based artificial agents/systems.

Coreference Resolution (CR) detects mentions and links the mentions referring to a common

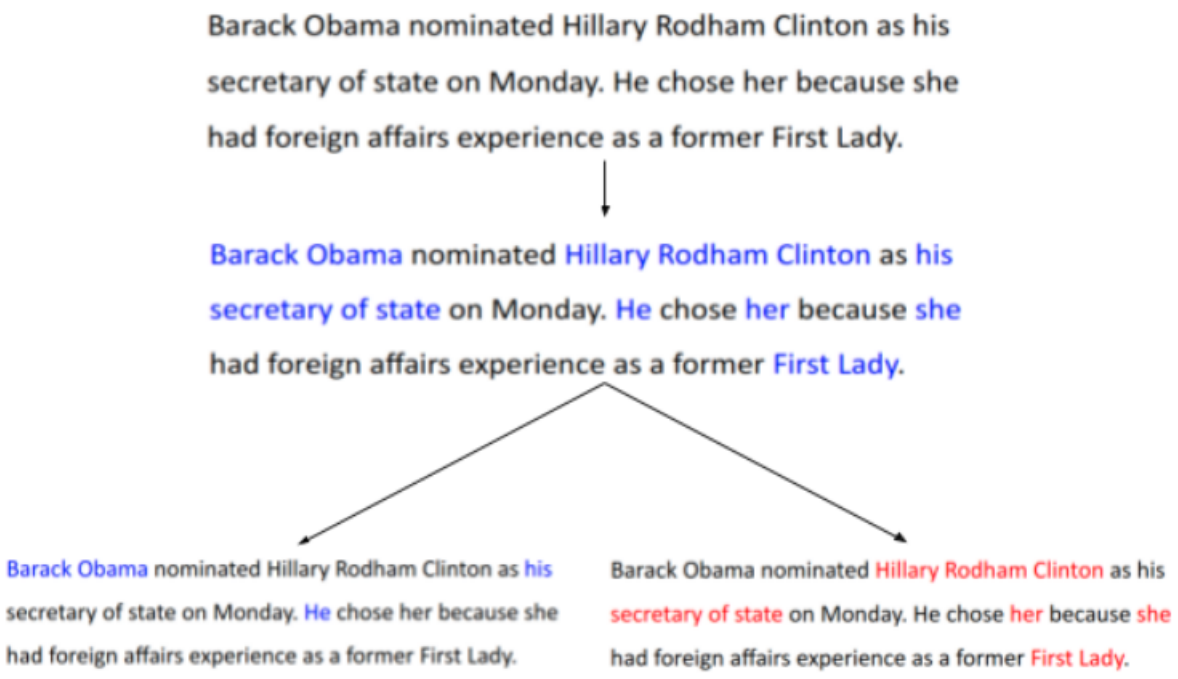

Figure 1.1: Mention detection and clustering in coreference resolution

entity. In figure 1.1 has two entities, *Barack Obama* and *Hilary Rodham Clinton*. There are two clusters of mentions {Barack Obama, He, his} and {Hillary Rodham Clinton, her, she, secretary of state, First Lady}. The first group of mentions refers to *Barack Obama*, and the second group of mentions refers to *Hillary Rodham Clinton*. These groups of mentions form coreference chains or mention clusters for the example passage. In this passage *Hillary Rodham Clinton* is an antecedent that appears before a referring mention *she*. An optimal CR system is expected to detect mentions {her, she, secretary of state, First Lady} and link them with a common entity *Hillary Rodham Clinton*.

In this example, suppose we want to answer the question: *Why Barack Obama nominated Hilary Rodham Clinton?* A correct linking of *She* with *Hilary Rodham Clinton*, *her* and *First Lady* gives clue about *foreign affairs experience* that leads to her nomination. It is evident from this discussion that CR is useful for other NLP tasks too.

1.2 Steps in Corefernce Resolution

Coreference resolution task is an assembly of two sub-tasks: mention detection and optimal clustering of mentions.

1.2.1 Mention Detection

Mention detection is the first stage of coreference resolution. The objective is to find the spans of text that constitute each mention. A mention can be a pronoun, noun, or name. NLP systems like named entity recognizer, part of speech tagger are useful for finding mentions. These NLP systems can able to find most instances of pronouns, noun phrases, or names correctly. However, all these instances may not be considered as mention. Mention detection algorithms are usually very liberal in proposing candidate mentions using such NLP tools. This approach creates a large candidate mentions space. Hence, the mentions space is needed to be pruned optimally for computational efficiency. Classifier model could be a better choice instead of these pipeline approach for mention detection. CR system computes mention score for each mention in the mentions space. It discards mention with a low score. Singleton mentions (mention with no antecedent) are not annotated in many datasets such as 2012 CoNLL shared task data [36].

1.2.2 Optimal Clustering of Mentions

Mentions with similar properties are grouped to build coreferent clusters in the second stage. Some linguistic properties considered among these mentions are number agreement, person agreement, gender or noun class agreement, binding theory constraint, recency, grammatical role, verb semantics, and selectional restriction.

CHAPTER 2

RELATED WORK

2.1 Coreference Resolution Models

2.1.1 Rule-based methods

Like other Natural Language Processing (NLP) tasks, earlier Coreference Resolution (CR) uses hand-crafted rules. Most earlier knowledge-rich algorithms are powered with hand-crafted rules that depend on semantic and syntactic features of text under consideration. Hobb's naive algorithm [21] was one of the first methodologies on anaphora resolution. It applies a rule-based left to right breadth-first traversal on the parse tree of a sentence to search and find an antecedent. It uses world knowledge-based selectional constraints for antecedent elimination. The rules and selectional constraints help the algorithm to converge to a single antecedent by pruning the antecedent search space. Lappin and Leass [26] proposed a hybrid algorithm. It considers syntax as well as a discourse for pronominal anaphora resolution. It has a discourse model that consists of all the potential antecedent references of a specific anaphor. Each antecedent has a salience value considering the semantic and syntactic constraints. The salience value of an antecedent depends on many other features. An antecedent with maximum salience value is considered as the best antecedent. This algorithm incorporates a signal attenuation mechanism that halves the influence or salience while propagating to the next sentence. BFP algorithm [5] is at the core of centering theory [16]. It uses discourse structure to explain phenomena such as anaphora and coreference. Hobb's dataset is used to evaluate the centering theory algorithm.

Rule-based algorithms have a high dependency on external knowledge. Efforts [2,17,20,27,47] were made to reduce the dependency of rules on external knowledge. Baldwin [2] proposed COGNIAC, a knowledge poor coreference resolver model with high precision. This model assumes there exists an anaphor subclass that doesn't need generic reasoning. This model can differentiate

between anaphor whether it needs external knowledge and or not. Attempts [19, 30] were made to incorporate world knowledge into a coreference resolution system. Haghighi and Klein [17] proposed a strong baseline to modularize syntactic, semantic and discourse constraints. This model outperformed supervised as well as unsupervised systems at that time.

2.1.2 Learning based

Availability of annotated corpora such as ACE [14], MUC [31] paved the path for learning based CR models. Even with this dataset Learning in coreference is an arduous task in NLP. Raghunathan et al. [38] demonstrated hand-engineered system built on top of parse trees had outperformed earlier learning-based approaches for coreference resolution. However, this approach overcame by the highly lexical learning method proposed by Durett and Klein [15]. Earlier neural models [7, 8, 45] archived better performance compared to machine learning-based models. All of these pipelined approaches use the mention proposal algorithm and rely on a parser to find head features. Parsing errors in this method causes cascading error in the overall model. First-time Daume III and Marcu [9] proposed a non-pipeline algorithm to jointly learn mention detection and coreference resolution. Learning-based coreference resolution systems can be broadly classified as: (1) mention-pair classifier, (2) entity-mention models, and (3) mention-ranking models

2.1.2.1 Mention Pair

Mention pair models [**?**,3,4,7,11,35,40,41,44] consider coreference as a collection of mention pair links. These models first detect mentions, then learn pairwise mention scores to classify, and finally cluster mention pairs. In the first step instances of valid mentions are detected. The algorithm proposed by Soon et al. [40] was a very popular mention creation algorithm. Subsequent mention creation algorithms apply constraints to minimize incorrect and remove hard-to-train mention instances. The second step trained a classifier to decide whether two mentions were co-referent or not. In the third step, used various clustering techniques to build a coreference chain.

Though mention-pair models have achieved popularity in CR task, it has some challenges to overcome. Issue of Transitivity constraint. If there were mention pairs (A, B) and (B, C) then it is expected to have a co-referent mention pair (A, C). The transitivity property should not be applied without considering other constraints. Let say He referred to Clinton and Clinton Referred to She then He and She should not be co-referent.

2.1.2.2 Entity-mention

Mention pair models are effective for the CR task, but they do not use entity-level information, i.e., features between clusters of mentions. Entity-level information is helpful to inform the new decision about the prior CR decision. Entity-mention CR systems use entity-level information. The procedure to find useful features itself was very challenging. So most works are done with mention pair modeling manner. Aggregated mention pair scores from these models can be useful to define entity-level features between clusters of mentions. Instead of mention and antecedent pair, entity models [6, 8, 18, 45] focus on past coreference decision to get utilized in new decisions. A mention is compared with a partially formed cluster instead of individual antecedents.

2.1.2.3 Mention and Span Ranking

The mention pair model acts as a binary classifier to decide whether two mentions are coreferent or not. However, the mention pair model provides no clue to compare to antecedent and select the optimal one for a given anaphor. This issue was handled by many of the ranking models [10, 15, 28, 29, 32, 39]. The ranking seems to be a better choice as compared to classification to handle CR task. Denis and Baldridge [11] used ranking loss function in place of classification. A ranking model proposed by Durrett and Klein [15] uses a log-linear model on surface features in antecedent selection. Even though popular, mention ranking models are not able to utilize the past decisions to make new decisions.

2.2 Error Analysis Coreference Resolution Systems

Coreference Resolution research has focused on the improvement of accuracy using evaluation metrics. These metrics provide a quantifiable summary of model performance on a pool of errors. Most performance analysis methods evaluate nominals, proper names, and pronouns separately. Methodologies focusing discussion on specific error or manual classification of a small set of errors, fail to quantify the impact of these errors. Holen [22] presented a manual analysis approach with a more comprehensive set of error types. It highlights evaluation metric shortcomings instead of model analysis. Stoyanov et al. [42] used gold annotation and evaluated improvement in mention detection, anaphoric mention detection, and named entity recognition. They defined nine resolution classes based on the mention types and antecedent properties. It can characterize the variation of resolution classes but missed out on cascade error when mentions resolved simultaneously. CoNLL shared task [36, 37] provides a multi-system comparison and measures the impact of mention and anaphoricity detection.

Kummerfeld and Klein [25] worked on detailed analysis of errors and extended earlier work on evaluation. Their method has a detailed understanding of error distribution instead of just accuracy comparison on evaluation metrics. Hence we consider analyzing errors analysis two current state-of-the-art systems [23, 24] using their methodology.

CHAPTER 3

EXPERIMENTAL RESULTS

3.1 Models

We use two state-of-the-art coreference resolution systems [23, 24] that use SpanBERT and BERT for embedding span representation. We consider the suggestion of Joshi et al. [24], to use independent versions of BERT-base [24] and SpanBERT-base [23]. The *base* variant of these models have lighter computational overhead as compared to *large* variants [13].

3.1.1 BERT-base Coreference System

Joshi et al. [24] used BERT transformer [13] based span embedding in place of LSTM based span embedding in earlier work of Lee et al. [29]. Originally BERT is pre-trained on BookCorpus and English Wikipedia with two training objectives: Masked Language Modeling (MLM) and next sentence prediction (NSP). BERT encoder generates contextualize vector representation of each input sequence of tokens.

$$P(y) = \frac{e^{s(x,y)}}{\sum_{y' \in Y} e^{s(x,y')}} \tag{3.1}$$

The model learns the distribution P(.) over possible antecedent spans Y for each mention span x. A scoring function s(x,y) uses mention scores of constituent spans and their joint compatibility score. The Span mention score tells how likely a span is a valid mention. Whereas compatibility score measures how likely two mentions refer to the same entity.

$$s(x, y) = s_m(x) + s_m(y) + s_c(x, y) \tag{3.2}$$

where:

Mention score of span x:

$$s_m(x) = FFNN_m(g_x) \tag{3.3}$$

Mention score of span y:

$$s_m(y) = FFNN_m(g_y) \tag{3.4}$$

Joint compatibility score of span x and y:

$$s_c(x, y) = FFNN_c(g_x, g_y, \phi(x, y)) \tag{3.5}$$

Spans x and y have span embeddings g_x and g_y, respectively. Speaker and distance information are categorised as meta-information $\phi(x, y)$. $FFNN_m$ and $FFNN_c$ represent Feed Forward Network in expressions to compute mention score and coreference score, respectively.

$$\log \prod_{i=1}^{N} \sum_{y' \in Y(i) \cap GOLD(i)}^{N} P(y') \tag{3.6}$$

The marginal log-likelihood of all correct antecedents for each span is optimized based on annotated gold clusters in the training data. The model selects the best antecedent on the basis of the calculated optimized score and forms the coreference chain retaining the transitivity property. In equation 3.6 GOLD(i) is the set of spans in gold clusters containing span i. This process accurately prunes spans and makes sure only gold mentions get positive updates.

3.1.2 SpanBERT-base Coreference System

Pre-trained SpanBERT by Joshi et al. [23] provides a better approach to represent and predict spans of text. Unlike BERT, SpanBERT masks a random contiguous span of tokens instead of individual tokens. They proposed Span Boundary Objective that able to predict the entire masked span using span boundary token representation. SpanBERT single sequences are used for training of encoder instead of bi-sequence unlike BERT with NSP objective. Apart from this modified embedding SpanBERT-base follows similar training and clustering as BERT-base.

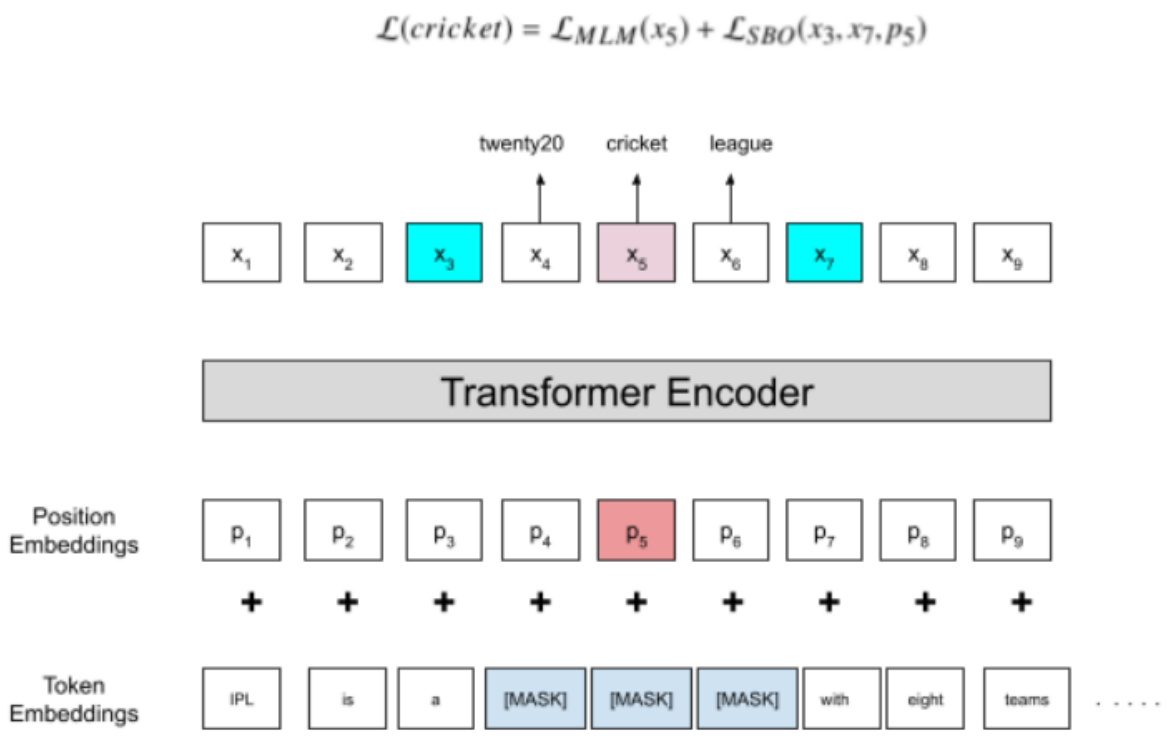

Figure 3.1: In this example the span *twenty20 cricket league* is masked. The Span Boundary Objective uses the boundary tokens output representation (x_3, x_7) and position embedding p_5 to predict the token (*cricket*) in masked span.

3.2 Dataset

We use the English portion of CONLL-2012 shared task data [36] for our experiments. This data set is most commonly used to evaluate many recent coreference models [23, 24, 28]. It is a document-level dataset with 3384 (2802 training, 343 development, and 348 test) documents having 1.6M words. Table 3.1 has information about the dataset. There are seven genres: broadcast conversations, broadcast news, magazine texts, news wire, pivot texts, telephone conversations, and weblogs. There are about one million words in this dataset. The annotation complexity for coreference increases non-linearly with the length of a document. Longer documents are split into parts to reduce annotation complexity. Three genres, telephone conversation, weblogs, and broadcast conversation, contribute to a large share of longer documents.

Section	Documents	Words	Mentions	Links	Entities
Total	3384	1.6M	194480	135179	44203
Train	2802	1.3M	155560	120417	35134
Validation	343	160K	19156	14610	4546
Test	348	170K	19764	15232	4532

Table 3.1: Number of documents, entities, links and mentions in the English part of OntoNotes v5.0 data [36]

3.3 Evaluation Metrics in Coreference Resolution

An evaluation metric for CR should consider two issues [33]: Interpretability and Discriminative power. High interpretability scores suggest the model is good at coreference relation detection. A highly discriminative model can differentiate a good decision from a bad one. There are several metrics proposed for the evaluation of CR. We consider three evaluation metrics commonly used for current research work and the 2012 ConLL shared task dataset [36] for our experiments. Each metrics has a separate dimension in focus. Links representation-based MUC [43], mention-based B-CUBED [1], and entity-based $CEAF$ [31].

3.3.1 MUC

Vilain et al. Vilain et al. [43] proposed the first-ever evaluation metric for coreference resolution task. The system predicted links are compared with manually annotated coreference chain or truth link. This metric computes the number of modifications needed to change a response set into a truth set. MUC precision and recall are calculated as follows.

$$MUCPrecision(G,P) = \sum_{p \in P} \frac{|p| - |partition(p,G)|}{|p| - 1} \tag{3.7}$$

In 3.7 |partition(p,G)| is the number of cluster in gold that predicted cluster intersects.

$$MUCRecall(G,P) = \sum_{g \in G} \frac{|g| - |partition(g,P)|}{|g| - 1} \tag{3.8}$$

In 3.16 |partition(g,P)| is the number of cluster in predicted that gold set intersects.

where partition $(x, y) = \{y|y \in Y \& y \in x \neq \phi\}$

This metric is the least discriminative compared to other subsequent metrics proposed for coreference resolution. It has a similar score for link joining singleton or most significant entities.

3.3.2 B^3 (B-Cubed)

Bagga and Baldwin [1] consider each individual mention to calculate precision and recall. The final number for precision and recall are computed as follows:

$$FinalPrecision = \sum_{i=1}^{N} w_i * Precision_i \tag{3.9}$$

$$FinalRecall = \sum_{i=1}^{N} w_i * Recall_i \tag{3.10}$$

In equation 3.9 and 3.10 and N = total number of entities in the document, and each entity i has an assigned weight w_i, precision as $Precision_i$ and recall as $Recall_i$. Weights to each entity i.e. $w_i = 1/N$.

3.3.3 CEAF

Constraint Entity Alignment F-measure (CEAF) by Luo [31] compares similarity between entity. The similarity measures create optimal mapping between predicted and truth clusters. This mapping is used to calculate precision and recall. There are four similarity measurements in this metric.

$\phi_1(G, P)$ considers two entities same if all mentions are same.

$$\phi_1(G, P) = \begin{cases} 1 & \text{if P=G} \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

$\phi_2(G, P)$ considers two entities same if there is at least a common mention.

$$\phi_2(G, P) = \begin{cases} 1 & \text{if P} \cap \text{G} \neq \phi \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

$\phi_3(G, P)$ counts the number of common mention between G and P

$$\phi_3(G, P) = |G \cap P| \tag{3.13}$$

F measure between G (gold entities) and P (predicted entities) is expressed as $\phi_4(G, P)$.

$$\phi_4(G, P) = 2.\frac{|G \cap P|}{|P| \cap |G|} \tag{3.14}$$

The function m(p) takes a predicted cluster p as input returns gold cluster g. A predicted cluster can only be mapped to one gold cluster. CEAF precision and recall is computed as follows:

$$CEAF_{\phi_i} Precision(G, P) = max_m \frac{\sum_{p \in P} \phi_i(p, m(p))}{\sum_{p \in P} \phi_i(p, p)} \tag{3.15}$$

$$CEAF_{\phi_i} Recall(G, P) = max_m \frac{\sum_{p \in P} \phi_i(p, m(p))}{\sum_{p \in P} \phi_i(g, g)} \tag{3.16}$$

3.3.4 CoNLL as official Score

$$CoNLL_{F1} = \frac{(B^3_{F1} + MUC_{F1} + CEAF_{F1})}{3} \tag{3.17}$$

The official score reported in CoNLL shared task 2012 by pradhan et al [36] is the unweighted average F1 scores from B^3, MUC and entity-based CEAF metrics (denoted as $CEAF_{\phi 4}$). However, a weighted average of these scores can be useful depending on a specific downstream task [12].

3.4 Experimental Setup and Results

3.4.1 Setup

We adapted the Pytorch implementation work [46] on Coreference Resolution using BERT and SpanBERT [23, 24]. These models are fine-tuned with document-level English data of OntoNotes 5.0 dataset [36]. We find documents are longer in this dataset. So multiple segments are used to read a complete document. We train Each model with documents having a different set of maximum segment lengths of 128, 256, 384, and 512. We randomly truncate longer documents to have eleven

segments to handle the issue of memory intense span representation. We consider models with a maximum segment length of 128 for our analysis work considering the finding of Joshi et al. [24]. We use batch sizes of one document, similar to Joshi et al. [24] and lee et al. [29]. We run both the model for 24 epochs with a dropout rate of 0.3. We conduct experiments on Nvidia TITAN RTX GPUs with 24GB memory. The average training time is around 4 hours for BERT-base and SpanBERT-base.

3.4.2 Results on Evaluation Metrics

	MUC			B^3			$CEAF_{\phi 4}$			Average
	P	R	F1	P	R	F1	P	R	F1	F1
BERT-base	83.41	78.65	80.96	73.98	68.39	71.07	71.44	64.66	67.88	73.30
SpanBERT-base	82.82	83.00	82.91	73.20	74.71	73.96	72.57	70.12	71.32	76.06

Table 3.2: Evaluation with an average F1 score of three metrics MUC, B and $CEAF_{\phi 4}$ on test dataset

We use the official CoNLL-2012 evaluation script [36] to report precision, recall and F1 for the three evaluation metrics MUC, B^3 and $CEAF_{\phi 4}$. We report coreference score as unweighted average F1 score of these metrics for our models. The SpanBERT-base system outperforms the BERT-base system in terms of MUC, B^3, $CEAF_{\phi 4}$, and the final average score. SpanBERT system has a higher coreference score due to the high recall and comparative precision to the BERT-base system.

CHAPTER 4

ERROR ANALYSIS

4.1 Error Classification

Evaluation metrics for Coreference Resolution (CR) consider overall model performance and have the conclusive notion that high-scoring models encounter fewer prediction errors. However, these metrics remain silent about types of error in each model. Kummerfeld and Klein [25] did an extensive error analysis of earlier CR models reported in CoNLL 2011 Shared task [37] and other publicly available models. It follows a two-step method having Transformation and Mapping to classify system prediction into seven categories of error.

4.1.1 Transformation

Firstly the system output is modified to gold annotation using a transformation process with the following five operations as demonstrated in Figure 4.1.

1. **Alter Span** modifies a predicted span into gold spans. *Alter Span* step shows *mention X* in the left-most entity is modified.

2. **Split** divides predicted entities to form gold entities. The left-most entity is divided into two entities in the *Split* step.

3. **Remove** deletes predicted mentions that are not part of gold entities. All X mention are deleted in *Remove* step.

4. **Introduce** creates singleton entities (with one mention) for every missing gold mention in system prediction. Three new mentions are created at the rightmost part during the *Introduce* step.

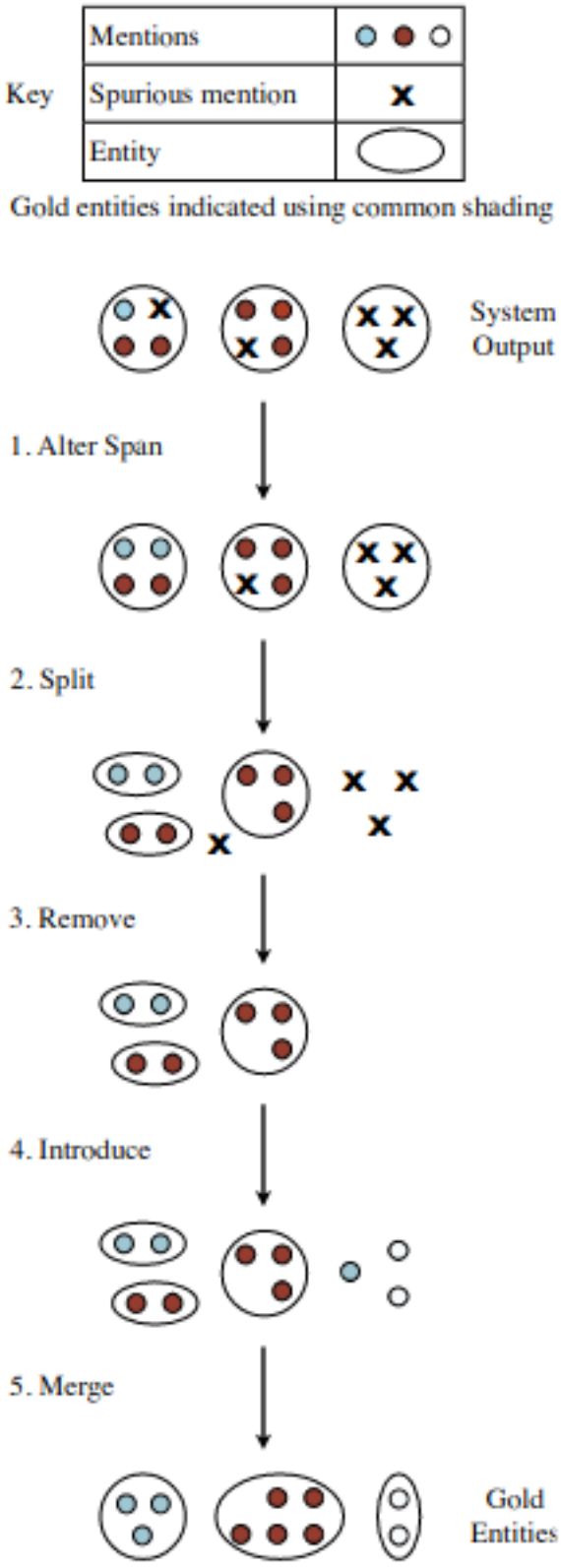

Figure 4.1: Transformation steps to change a predicted output (top) into a gold annotation (at buttom). Figure after Kummerfeld and Klein [25]

5. **Merge** unites a group of wrongly predicted entities to form one correct gold entity. Similar mention grouped and formed three entities at the end of *Merge* step.

4.1.2 Mapping

Secondly, these transformations contribute to seven types of errors: namely, i. Span Error, ii. Missing Entities, iii. Extra Entities, iv. Missing Mentions, v. Extra Mentions, vi. Divided Entities, and vii. Conflated Entities. We discuss each error in the Discussion on Error Analysis.

4.2 Discussion on Error Analysis

In this section, we compare the results of BERT-base and SpanBERT-base based Coreference Resolution systems. We consider seven categories of errors adapted from the work of Kummerfeld and Klein [25]. They evaluated earlier Coreference models on CoNLL-2011 shared task data [37]. The shared task reported the performance of many CR systems of that time. Our models use CoNLL-2012 [36] data. The English portion in CoNLL-2012 shared task data [37] has 1.3M words compared to 1M words in CoNLL-2011 shared task data [36]. Earlier work of Bjorkelund shows the addition of 160K words in evaluation data for training failed to improve the model performance [36] compared to models trained on only training data. It will be unfair to directly compare earlier models reported in their [25] analysis. However, it can give some clue about how current systems behaves.

System	Mention Detection	MUC	B^3	$CEAF_{\phi 4}$	Span Errors	Conflated Entities	Extra Mentions	Extra Entities	Divided Entities	Missing Mentions	Missing Entities
BERT-base	85.45	80.96	71.07	67.88	256	1103	507	406	1286	735	813
SpanBERT-base	86.85	82.91	73.96	71.32	272	1048	653	522	1086	589	558
BERKELEY [25]	75.57	66.43	66.17	NA	392	1694	923	833	1981	899	801

Table 4.1: Counts for each error type for BERT-base and SpanBERT-base on the English test set of the 2012 CoNLL shared task and the Best performing model BERKELEY on 2011 CoNLL shared task reported by Kummerfeld and Klein [25]

Error	System	Gold
Extra text	*Judy Miller as a journalist*	*Judy Miller*
Missing text	*them*	*them all*

Table 4.2: Examples of Span errors with Extra text and Missing text

4.2.1 Span Errors

Span errors occur due to missing or extra text in spans. A missing text is present in gold mentions but absent in system predicted mentions. Whereas extra text is present in system predicted spans but absent in gold spans.

	BERT-base		SpanBERT-base	
Type	Extra	Missing	Extra	Missing
NP	4	1	6	4
POS	121	5	120	5
.	1	7	3	3
SBAR	6	0	4	0
PP	16	0	16	2
DT	30	3	31	4

Table 4.3: Counts of Span Errors grouped by the labels (NP: Noun Phrase, POS: Possessive Ending (e.g. people's, government's) , .: Punctuation, SBAR: Subordinate clause, PP: Prepositional phrase, DT: Determiner) over the extra/missing part of the mention.

Table 4.3 shows parse nodes having only missing and extra text in gold parse. It shows in both models have more Extra text cases for Span Error. The *POS: possessive* type parse node witnesses maximum differences in missing and extra for both the model, seems superficial. Span errors can be minimized by reducing parsing-related errors [25]. It is a challenging task to completely remove the parsing issue because of inconsistency in the annotation of the dataset.

4.2.2 Extra Mentions and Missing Mentions

Table 4.4 shows antagonistic behavior of Extra and Missing Mentions errors. If a predicted entity has more mentions than a gold entity causes an Extra Mentions error. In the case of Missing Mentions, the error system predicted entity has less mention than a gold entity.

Error	System	Gold
Extra Mention	*Focus Today*	Focus Today
	we	-
	us	-
	our	-
	our program	our program
Missing Mention	this SMS	*this SMS*
	it	*it*
	-	*this*

Table 4.4: Examples of Extra Mentions and Missing Mentions error.

	BERT-base		SpanBERT-base		
Mention	Extra	Missing	Extra	Missing	Count
Proper Name	122	171	125	149	
Nominal	229	334	290	288	
Pronoun	156	230	238	152	
it	32	26	46	24	1318
you	16	54	51	25	1273
we	17	48	29	25	754
us	5	6	6	2	265
that	6	7	7	7	2209
they	6	13	7	9	939
their	10	7	10	2	457
this	7	12	11	13	989

Table 4.5: Counts of Missing and Extra Mentions errors by mention type, and some of the common mentions.

	BERT-base				SpanBERT-base			
	Proper Name		Nominal		Proper Name		Nominal	
	Extra	Missing	Extra	Missing	Extra	Missing	Extra	Missing
Text_Match	60	76	53	69	60	73	63	61
Head_Match	82	123	133	184	88	110	161	150
Others	40	48	96	150	37	39	128	138
NER Matches	66	74	19	13	68	64	11	12
NER Differs	3	3	0	0	2	2	1	0
NER Unknown	53	94	210	321	55	83	278	276
Total	122	171	229	334	125	149	290	288

Table 4.6: Counts of Extra and Missing Mentions, grouped by properties of the mention and the entity it is in

Table 4.5 lists Missing and Extra errors by type of mentions involved. It also lists some of the commonly occurring Missing and Extra mentions. BERT-base system with high precision has few Extra and more Missing mentions. Whereas the SpanBERT-base system with a high recall has more Extra and fewer Missing mentions.The mentions *you* and *we* occur most frequently in Missing error for BERT-base and Extra error for SpanBERT-base. We observe Missing mentions are penalized highly in SpanBERT-base as compared to BERT-base. We group Extra Mentions and Missing Mentions errors by proper names and nominals in Table 4.6. The first section of the table reports errors that consider the exact string match or head match between the mentions with error and the mentions in the cluster. Named entity annotation of mention with error is considered, in the second section. It measures occurrences of matched mention type with that of cluster type. There are balanced occurrences of two types of errors in all cases. However, one exception is observed for unknown NER for nominal in the BERT-base model. Models included in earlier work [25] reported this exception for exact string match case for nominal. We differ from some of the earlier observations. Our models can identify pleonastic pronouns more effectively than the models reported in the work of [25]. BERT-base shows better performance for Extra error whereas SpanBERT-base is better for Missing error, both concerning instances with head matching.

4.2.3 Extra Entities and Missing Entities

An entity is a set of all mentions having the same references. A missing Entity is a gold entity, which is not predicted by the system. An extra entity is introduced by the system, which is not a gold entity. These two cases contribute to Missing Entities and Extra Entities errors.

Error	System	Gold
Extra Entity	*Dear viewers*	-
	dear viewers	-
Missing Entity	-	*everyone*
	-	*you*

Table 4.7: Examples of Extra Entities and Missing Entities error.

Composition			BERT-base		SpanBERT-base	
Name	Nominals	Pronoun	Extra	Missing	Extra	Missing
0	1	1	84	212	101	156
1	0	1	8	15	11	10
1	1	0	23	58	26	45
2	0	0	39	56	46	35
0	2	0	153	302	216	215
0	0	2	38	16	43	9
3+	0	0	3	8	4	5
0	3+	0	20	48	43	29
0	0	3+	17	22	15	9
		Others	21	76	17	45
		Total	406	813	522	558

Table 4.8: Counts of Extra and Missing Entity errors,grouped by the composition of the entity (Names, Nominals, Pronouns).

Table 4.8 reports these two errors considering the composition (name, nominal, and pronoun) of entities. A noticeable difference is observed for these two errors. Entities containing one nominal and one pronoun (row 0 1 1) have more Missing errors than Extra errors. Entities with two pronouns (row 0 0 2) behave oppositely, having more Extra errors compared to Missing errors. SpanBERT-base has more Extra error and less Missing error for entities with three pronoun (row 0 0 3+) or three nominals (row 0 3+ 0), whereas BERT-base shows an opposite trend. Single type error contributes for 66.50% Extra Entity errors and 55.59% Missing Entity error in Bert-base and 70.30% Extra Entity errors and 54.12% Missing entity errors in SpanBERT-base model. These results show entities with a single type mention contribute the most for these two errors.

Table 4.9 presents entity errors with a single type mention and categorized as three groups: Exact, Head, and Non. Nominals account for maximum occurrences and variation across these categories for both SpanBERT-base as well as BERT-base. Head match constitutes about half the nominal for Extra column as well as Missing column. The higher share of Head match suggests these neural models are good at head finding in the mention. Table 4.8 reports entity containing a pronoun and a nominal comes as the second most error case. Table 4.10 presents the list of most frequent pronouns for errors with a pronoun and name or nominal.

We can consider pronouns as a reference to interpret these errors. A pronoun can be an

Match	Type	BERT-base		SpanBERT-base	
		Extra	Missing	Extra	Missing
Exact	Proper Name	22	26	26	15
	Nominal	67	71	91	49
	Pronoun	36	13	41	8
Head	Proper Name	32	46	40	25
	Nominal	127	188	173	130
	Pronoun	36	13	41	8
Non	Proper Name	10	18	10	15
	Nominal	46	162	86	114
	Pronoun	19	25	17	10

Table 4.9: Counts of Extra and Missing Entity errors grouped by properties of the mentions in the entity.

Mention	BERT-base		SpanBERT-base	
	Extra	Missing	Extra	Missing
that	27	72	31	59
it	23	59	33	40
this	14	35	16	27
they	6	12	7	4
their	2	8	2	4
them	1	7	1	6
Any pronoun	92	227	112	164

Table 4.10: Counts of common Missing and Extra Entity errors where the entity has just two mentions: a pronoun and either a nominal or a proper name.

Extra mention predicted incorrectly as coreferent or a Missing mention predicted incorrectly as non-coreferent. Table 4.5 shows these errors are biased towards Missing in BERT-base whereas SpanBERT-base biased towards Extra errors. However, the distribution of these errors speaks differently. For example *that* is balanced for both the errors in Table 4.5 whereas in Table 4.10 biased towards Missing Entity Error. Entities that have either a nominal or a nominal with a pronoun, dominates Extra entity and Missing Entity errors. We report head matching in these cases is quite misleading. Kummerfeld and Klein [25] reported String match as misleading. This suggests the use of semantics, context, and discourse to reduce these two errors.

4.2.4 Conflated Entities and Divided Entities

Table 4.11 lists Conflated Entity error: mentions in separate gold entities are predicted as a single entity. Divided Entity error: mentions in one gold entity are predicted as separate entities.

Error	System	Gold
Conflated Entity	*the anti phased motion*$_1$	the anti phased motion$_1$
	this$_1$	this$_2$
	it$_1$	it$_2$
	they$_1$	they$_1$
Divided Entity	*the two of you*$_1$	*the two of you*$_1$
	the two honorable guests$_2$	*the two honorable guests*$_1$
	both of you$_1$	*both of you*$_1$
	two honorable guests$_2$	*two honorable*$_1$ *guests*

Table 4.11: Examples of Conflated Error and Divided error.

Incorrect Part			Rest entity			BERT-base		SpanBERT-base	
Name	Nominal	Pronoun	Name	Nominal	Pronoun	Conflated	Divided	Conflated	Divided
0	0	1+	0	0	1+	89	72	70	43
0	0	1+	0	1+	1+	110	105	112	87
0	0	1+	0	1+	0	181	231	182	228
0	1+	0	0	1+	0	128	138	145	156
0	0	1+	1+	1+	1+	74	98	69	80
0	0	1+	1+	0	1+	67	111	67	71
0	0	1+	1+	0	0	89	61	27	50
Others						365	470	376	371
Total						1103	1286	1048	1086

Table 4.12: Counts of Conflated and Divided entities errors grouped by the Name /Nominal/Pronoun composition of the parts involved.

Table 4.12 lists Conflated Entities and Divided Entity errors as per the composition of part split/merged and the rest of the entity. 1+ / 0 depicts the count of each type of mention in the entity. Misplacement of pronouns constitutes the largest portion of these errors. The most common errors involve parts with just pronouns. The issue becomes challenging not to have a proper name in the remaining part of the entity. Systems may have conflated pronouns of two entities together to creates this core issue of entities entirely having pronouns.

Aggregating instances of the incorrect part containing a single pronoun in Table 4.12: It accounts for 42.33% and 41.60% of conflated cases for BERT-base and SpanBERT-base; 39.81%

and 41.25% of divided cases for BERT-base and SpanBERT-base. There is a good possibility of cases when a part is both conflated with a wrong entity and divided from its true entity. If a pronoun is placed in the wrong entity causes a Pronoun link error. Table 4.12 shows Pronoun link error is very common in Conflated Entities and Divided Entities.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this thesis, we evaluate the performance of two end-to-end Coreference Resolution (CR) systems BERT-base [24] and SpanBERT-base [23] on CoNLL-2012 Shared Task data [36]. We report their performance as the unweighted average F1 scores: 73.30 for BERT-base (higher precision) and 76.06 SpanBERT-base (higher recall). We further investigate the error distributions of both the systems based on the work of Kummerfeld and Klein [25]. We observe the same model is not outperforming in all error types. The BERT-base has more errors for Missing Mentions, Missing Entities, and Divided Entities. The SpanBERT-base has more errors in Span Errors, Extra Mentions, Extra Entities, and Conflated Entities. We observe SpanBERT handles recall-related issues better than BERT-base and systems reported by Kummerfeld and Klein [25].

Considering the patterns in Span errors, It seems an optimal parsing method can be helpful to minimize this error. We report nominals to contribute to maximum Missing and Extra Mentions errors. The nominals in the dataset also have nested annotation, which could lead to a mismatch. Text match cases witness a balanced distribution of Extra and Missing mention errors. Our analysis suggests more information needs to be included even though span Head matches. The composition of entities is crucial in the case of the Extra Entity and Missing Entity errors. Entities having one type of component has a maximum share in these errors. Among single type error nominals with head-match contribute to the maximum across the composition. We also report pronoun contributes to a large portion of error distribution in Conflated and Divided Entities. Pronoun grouped in the wrong mentions cluster of an entity causes a cascaded pronoun linking error. Accurate linking of the pronoun with an entity is desirable in this task.

Downstream NLP tasks such as Question Answering or Text Summarization can achieve better performance by resolving references [34]. The reference requirement changes as per the need of the respective task. Our analysis work will be helpful to select an optimal CR model considering the objective of a downstream NLP task.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer, 1998.

[2] Breck Baldwin. CogNIAC: high precision coreference with limited knowledge and linguistic resources. In *Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, 1997.

[3] Eric Bengtson and Dan Roth. Understanding the value of features for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 294–303. Association for Computational Linguistics, 2008.

[4] Anders Björkelund and Richárd Farkas. Data-driven multilingual coreference resolution using resolver stacking. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 49–55. Association for Computational Linguistics, 2012.

[5] Susan E. Brennan, Marilyn W. Friedman, and Carl J. Pollard. A centering approach to pronouns. In *Proceedings of the 25th Annual Meeting on Association for Computational Linguistics*, page 155–162. Association for Computational Linguistics, 1987.

[6] Kevin Clark and Christopher D. Manning. 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. Association for Computational Linguistics, 2015.

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

[8] Kevin Clark and Christopher D. Manning. 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. Association for Computational Linguistics, 2016.

[9] Hal Daumé III and Daniel Marcu. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 97–104. Association for Computational Linguistics, 2005.

[10] Pascal Denis and Jason Baldridge. A ranking approach to pronoun resolution. In *IJCAI*, volume 158821593, 2007.

[11] Pascal Denis and Jason Baldridge. Specialized models and ranking for coreference resolution. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 660–669, 2008.

[12] Pascal Denis and Jason Baldridge. Global joint models for coreference resolution and named entity classification. *Procesamiento del lenguaje natural*, 42, 2009.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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. Association for Computational Linguistics, 2019.

[14] George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon, 2004.

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

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

[17] Aria Haghighi and Dan Klein. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1152–1161, 2009.

[18] Aria Haghighi and Dan Klein. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 385–393, 2010.

[19] Sanda Harabagiu and Steven J Maiorano. Knowledge-lean coreference resolution and its relation to textual cohesion and coherence. In *The Relation of Discourse/Dialogue Structure and Reference*, 1999.

[20] Sanda M. Harabagiu, Razvan C. Bunescu, and Steven J. Maiorano. Text and knowledge mining for coreference resolution. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*, 2001.

[21] Jerry R. Hobbs. Resolving pronoun references. *Lingua*, 44(4):311 – 338, 1978.

[22] Gordana Ilić Holen. Critical reflections on evaluation practices in coreference resolution. In *Proceedings of the 2013 NAACL HLT Student Research Workshop*, pages 1–7, Atlanta, Georgia, 2013. Association for Computational Linguistics.

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

[24] Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5802–5807. Association for Computational Linguistics, 2019.

[25] Jonathan K. Kummerfeld and Dan Klein. Error-driven analysis of challenges in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 265–277, Seattle, Washington, USA, 2013. Association for Computational Linguistics.

[26] Shalom Lappin and Herbert J. Leass. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561, 1994.

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

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

[29] Kenton Lee, Luheng He, and Luke Zettlemoyer. 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. Association for Computational Linguistics, 2018.

[30] Tyne Liang and Dian-Song Wu. Automatic pronominal anaphora resolution in english texts. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 9, Number 1, February 2004: Special Issue on Selected Papers from ROCLING XV*, pages 21–40, 2004.

[31] Xiaoqiang Luo. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, 2005.

[32] Sebastian Martschat and Michael Strube. Latent structures for coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:405–418, 2015.

[33] Nafise Sadat Moosavi and Michael Strube. Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–642. Association for Computational Linguistics, 2016.

[34] Thomas S. Morton. Using coreference for question answering. In *Coreference and Its Applications*, 1999.

[35] Vincent Ng and Claire Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111. Association for Computational Linguistics, 2002.

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

[37] Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. CoNLL-2011 shared task: Modeling unrestricted coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27, Portland, Oregon, USA, 2011. Association for Computational Linguistics.

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

[39] Altaf Rahman and Vincent Ng. Supervised models for coreference resolution. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pages 968–977, 2009.

[40] Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001.

[41] Veselin Stoyanov, Claire Cardie, Nathan Gilbert, E. Riloff, David J. Buttler, and D. Hysom. Reconcile: A coreference resolution research platform. 2010.

[42] Veselin Stoyanov, Nathan Gilbert, Claire Cardie, and Ellen Riloff. Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 656–664. Association for Computational Linguistics, 2009.

[43] Marc Vilain, John D Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, 1995*, 1995.

[44] Sam Wiseman, Alexander M. Rush, Stuart Shieber, and Jason Weston. 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. Association for Computational Linguistics, 2015.

[45] Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 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. Association for Computational Linguistics, 2016.

[46] Liyan Xu and Jinho D. Choi. Revealing the myth of higher-order inference in coreference resolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8527–8533. Association for Computational Linguistics, 2020.

[47] Amir Zeldes and Shuo Zhang. When annotation schemes change rules help: A configurable approach to coreference resolution beyond ontonotes. In *Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2016)*, pages 92–101, 2016.