Conundrums in Entity Coreference Resolution: Making Sense of the State of the Art

Jing Lu and **Vincent Ng**
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688
{ljwinnie,vince}@hlt.utdallas.edu

Abstract

Despite the significant progress on entity coreference resolution observed in recent years, there is a general lack of understanding of what has been improved. We present an empirical analysis of state-of-the-art resolvers with the goal of providing the general NLP audience with a better understanding of the state of the art and coreference researchers with directions for future research.

1 Introduction

The advent of the neural NLP era has revolutionized virtually all areas of NLP research. For entity coreference, many issues that were once thought to be important no longer appear to be particularly relevant to the current research agenda. Specifically, while a decade ago coreference researchers have focused on developing computational models that are complex (e.g., structured models (Fernandes et al., 2012; Björkelund and Kuhn, 2014; Martschat and Strube, 2015)) and knowledge-rich (e.g., those that encode world knowledge (Ponzetto and Strube, 2007; Rahman and Ng, 2011a; Hajishirzi et al., 2013)), nowadays virtually all state-of-the-art resolvers employ a simple model (i.e., the mention-ranking model, which was developed more than a decade ago (Denis and Baldridge, 2008))[1] and a fairly simple input representation (i.e., contextualized word embeddings) in conjunction with a mechanism for learning representations of entity mention spans such that coreferent mentions have similar representations (Lee et al., 2017, 2018; Kantor and Globerson, 2019; Joshi et al., 2019).

Despite significant progress in the past few years in terms of performance numbers, what seems to be missing is an understanding of what has been improved. The lack of understanding has long been a concern shared by coreference researchers, even before the neural revolution in NLP. This has led to several attempts to analyze coreference resolvers over the years (Stoyanov et al., 2009; Kummerfeld and Klein, 2013). With the development of neural resolvers, however, this concern has become more serious than ever: the fact that significant progress can be made via learning mention representations with a simple neural mention-ranking model that employs a fairly simple input representation for a task as challenging as coreference resolution (CR) is somewhat contrary to common wisdom.

In light of this apparent conundrum, we present an empirical analysis of state-of-the-art entity coreference resolvers through four major sets of experiments in this paper, with the goal of gaining insights into their behaviors. We believe that our analysis will not only provide the general NLP audience with a better understanding of the state of the art, but also provide coreference researchers with directions for future research.

2 Evaluation Setup

In this section, we describe the datasets, the evaluation metrics, the state-of-the-art resolvers and the hyperparameters used in our experiments.

Datasets. We report results on three coreference datasets. The NIST-sponsored ACE evaluations resulted in several datasets. We use ACE 2005 (Walker et al., 2006), the last one in the series. The ACE 2005 organizers have only made the official training set (but not the official test set) publicly available, so previous work defined different train-test splits over the official training set. We employ the same train-test split as Bansal and Klein (2012).

KBP is another series of NIST-sponsored evaluations in the mid 2010s. KBP does not have any evaluations on entity CR, but to support high-level

[1] The first learning-based resolver is a pairwise ranker (Connolly et al., 1994), which was extended by Denis and Baldridge (2008) to rank more than two candidate antecedents at a time.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 6620–6631,
November 16–20, 2020.

	ACE			OntoNotes			KBP		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
#docs	365	117	117	2802	343	348	360	97	168
#mentions	34481	11126	9261	155558	19155	19764	40628	10983	13860
#chains	11963	3798	3050	35142	4545	4532	14332	3942	5482

Table 1: Dataset statistics in terms of the number of documents, mentions, and coreference chains.

information extraction tasks (e.g., event extraction, event CR), the organizers have made available several corpora that include entity coreference annotations. For training, we use three such corpora (LDC2015E29, LDC2015E68, and LDC2016E64). For evaluation, the KBP 2017 organizers have made available the official test set for the event CR task (LDC2017E51), which also include entity coreference annotations. We use it as our test set.

OntoNotes (Hovy et al., 2006), which was developed circa 2006, is the most widely-used dataset for entity coreference evaluations. It has a standard train-dev-test split. Unlike in ACE and KBP, singleton clusters are not annotated in OntoNotes.

The key difference between these three corpora is that OntoNotes supports "unrestricted" CR, meaning that coreference links are annotated between entity mentions without regard to their entity types. In contrast, coreference links are only annotated between mentions belonging to one of the seven entity types in ACE and one of the five entity types in KBP. Statistics on these corpora are shown in Table 1.

Evaluation metrics. Following the convention established in the CoNLL 2011 and 2012 shared tasks (Pradhan et al., 2011, 2012), we use as our primary coreference evaluation measure the CoNLL score, which is the unweighted average of the F-scores provided by three popular metrics, the link-based MUC metric (Vilain et al., 1995), the mention-based B^3 metric (Bagga and Baldwin, 1998), and the entity-based $CEAF_e$ metric (Luo, 2005). We obtain these scores using the official CoNLL scorer (Pradhan et al., 2014).[2]

Mention detection (MD) is the task of extracting the mentions in a text needed for entity CR. A key observation made in the CoNLL shared tasks was that the performance of resolvers was limited by MD, so it is important to examine the extent to which MD performance has improved over the years. We report performance in terms of recall, precision, and F-score, considering that a system mention is correctly detected if and only if it has an exact match in boundary with a gold mention.

Systems. We evaluate five variants of three state-of-the-art neural resolvers, all of which employ a ranking model where all candidate antecedents are ranked against each other for a given anaphor.

The first resolver, the Stanford neural resolver (Clark and Manning, 2016)[3], takes as input a set of entity mentions identified for a given document by a rule-based MD system and trains using reinforcement learning a simple mention ranker consisting of three hidden layers of ReLU units and a final layer that is fully-connected.

The other two resolvers are developed by Lee et al. (2018)[4] and Joshi et al. (2019)[5]. Both are *span-based* models, which have two key characteristics. First, mention spans are identified as part of CR, so this mitigates the propagation of errors from MD to CR. Second, *representations* of entity mention spans are learned so that coreferent mentions have similar representations. The key differences between these resolvers are: (1) in Lee et al. the input instances correspond to the sentences in the given document, whereas in Joshi et al. the input instances correspond to fixed-length non-overlapping segments of the input document[6]; (2) Lee et al. use a LSTM, whereas Joshi et al. use a transformer; and (3) the pretrained embeddings are different.

For each of these resolvers, we derive two variants. Specifically, Lee et al. (2018) employ GloVe+ELMo embeddings, but to better understand the effect of the contextual information provided by ELMo embeddings (Peters et al., 2018) on CR performance, we evaluate a version of Lee et al. using only GloVe embeddings (Pennington et al., 2014). We will henceforth refer to these two versions of Lee et al. as ELMo (i.e., ELMo+GloVe) and GloVe, respectively. Joshi

[2]LEA (Moosavi and Strube, 2016) is a coreference evaluation metric recently designed to address the shortcomings associated with B^3 and $CEAF_e$, but we found no difference in the performance trends in our experiments according to CoNLL and LEA. See the Appendix for the LEA results.

[3]https://github.com/clarkkev/deep-coref

[4]https://github.com/kentonl/e2e-coref

[5]https://github.com/mandarjoshi90/coref

[6]This is the *independent* version in Joshi et al. (2019).

Hyperparameter	ACE			OntoNotes			KBP		
	GloVe/ELMo	SpanB-b	SpanB-l	GloVe/ELMo	SpanB-b	SpanB-l	GloVe/ELMo	SpanB-b	SpanB-l
Max span width	30	30	30	30	30	30	20	20	10
Max top antecedents	50	50	50	50	50	50	50	50	35
Max training segs/sents	50 sents	3 segs	3 segs	50 sents	3 segs	3 segs	50 sents	3 segs	3 segs
Top span ratio	0.4	0.35	0.4	0.4	0.4	0.4	0.35	0.35	0.35
Max segment length	–	384	512	–	384	512	–	384	512
SpanBERT learning rate	–	2e-5	1e-5	–	2e-5	1e-5	–	2e-5	2e-5
Task learning rate	0.001	1e-4	3e-4	0.001	1e-4	3e-4	0.001	1e-4	2e-4

Table 2: Best hyperparameters obtained on the development sets for each span-based resolver.

et al. (2019) employ embeddings pretrained using a new method called SpanBERT (Joshi et al., 2020), which is designed to better represent text *spans* than BERT. The two variants of Joshi et al. differ in the transformer. Specifically, SpanBERT-base (henceforth SpanBERT-b) employs a simple transformer while SpanBERT-large (henceforth SpanBERT-l) employs a more complex transformer.

We use the publicly-available implementation of each of these resolvers. There is one caveat, however. Recall that the span-based resolvers were all evaluated on OntoNotes. Since singleton clusters are not annotated in OntoNotes, all singleton clusters predicted by a resolver are removed from its output before it is sent to the scoring program. In contrast, singleton clusters that contain mentions belonging to one of the ACE/KBP entity types are annotated in ACE/KBP, so these mentions should not be removed from a resolver's output. However, span-based resolvers cannot distinguish between spans that correspond to entity mentions and those that do not. To address this problem, we extend the span-based models so that they are jointly trained to predict entity mention spans and coreference links. Specifically, the feedforward neural network that is responsible for scoring a span in these models currently do not receive *direct* feedback on whether a span corresponds to an entity mention. We first turn it into a mention detector by training it in a supervised manner using the negative cross entropy loss, so that it predicts a positive mention score for a span if and only if the span corresponds to an entity mention. Then, to jointly learn MD and CR in the span-based resolvers, we employ a loss function that is the unweighted sum of the coreference loss and the MD loss.

Hyperparameter tuning. To ensure a fair comparison of the resolvers, we tune their hyperparameters to maximize the CoNLL score on development data. Note, however, that the authors of Stanford, ELMo, SpanBERT-b, and SpanBERT-l reported the best hyperparameter settings on OntoNotes in the original papers (Lee et al., 2018; Joshi et al., 2019), so we simply use them in our experiments and focus on tuning the hyperparameters for the remaining cases. We adopt the set of hyperparameters to be tuned from the original papers.

For Stanford, there are three hyperparameters to tune: α_{WL}, α_{FA}, and α_{FN}. These are the weights associated with three different types of mistakes made by the coreference model. Following Clark and Manning (2016), we fix α_{WL} = 1.0 and search for α_{FA} and α_{FN} out of {0.1, 0.2, . . ., 1.5} using a variant of grid search. For ACE, (α_{WL}, α_{FA}, α_{FN}) = (1.0, 0.5, 1.0) is the best configuration, and for KBP, the best configuration is (1.0, 0.5, 0.8). For OntoNotes, we use the configuration found by Clark and Manning, which is (1.0, 0.5, 0.8).

For GloVe and ELMo, we have five hyperparameters to tune. Specifically, we search for: (1) max span width (i.e., maximum number of words in a candidate span) out of {10, 20, 30}; (2) max top antecedents (i.e., maximum number of candidate antecedents) out of {35, 40, 45, 50}; (3) max training sentences out of {25, 50, 75, 100}; (4) task learning rate out of {5e-4, 1e-3, 2e-3}; and (5) top span ratio (i.e., the fraction of top spans that survive the filtering) out of {0.3, 0.35, 0.4, 0.45, 0.5}. For the two SpanBERT resolvers, we have seven hyperparameters to tune. For three of the hyperparameters (max span width, max top antecedents, and top span ratio), the ranges are the same as those used in GloVE and ELMo. For the remaining four, we search for: (1) max training segments out of {3, 4, 5}; (2) max segment length out of {128, 256, 384, 512}; (3) SpanBERT learning rate out of {1e-5, 2e-5}; and (4) task learning rate out of {1e-4, 2e-4, 3e-4}. Table 2 shows the best hyperparameter setting of each span-based model on each dataset.

3 Performance across Datasets

We first provide the reader with a high-level understanding of the state of the art by analyzing the five

		ACE				OntoNotes				KBP			
	System	CoNLL	MUC	Sing.	MD	CoNLL	MUC	Sing.	MD	CoNLL	MUC	Sing.	MD
1	Stanford	38.6	57.9	56.8	50.6	65.5	74.3	–	80.1	26.2	48.8	42.8	26.4
2	GloVe	68.0	76.6	67.0	87.4	68.1	76.7	–	82.0	65.9	71.3	63.2	82.4
3	ELMo	71.8	79.5	70.1	90.4	73.0	80.5	–	85.1	69.7	73.8	67.5	85.3
4	SpanBERT-b	75.7	83.1	71.5	91.6	77.4	83.7	–	87.1	71.3	75.6	67.5	86.3
5	SpanBERT-l	78.9	85.4	75.1	92.3	79.6	85.3	–	88.2	75.8	80.1	71.1	88.5

Table 3: Results of the resolvers on the three coreference datasets.

resolvers' performance on the three datasets.

Performance across datasets. Results on the three datasets, which are reported in terms of the CoNLL score, are shown in the CoNLL column in Table 3.[7] Although the five resolvers have been evaluated solely on OntoNotes, their relative performances are consistent across the datasets. In particular, the use of ELMo embeddings enables ELMo to outperform GloVe by 3.8–4.9% points. SpanBERT-b outperforms ELMo by 1.6–4.4% points, and SpanBERT-l further outperforms SpanBERT-b by 2.2–4.5% points.

Source of performance improvements. Do the above improvements stem from improved recognition of coreference links, or improved recognition of singleton clusters, or both? To understand whether these resolvers have improved in terms of *link* prediction, we examine the MUC F-scores (see the MUC column), which are computed solely on coreference links. As we can see, the MUC scores are consistently increasing down the table across all datasets, meaning that later systems are indeed doing better at identifying coreference links. To understand whether later resolvers are also better at identifying singleton clusters, we show in the Singleton column the percentage of singleton clusters that are correctly *recalled*. Again, the scores are increasing down the table, and the degree of improvement is particularly large from GloVe to ELMo and from SpanBERT-b to SpanBERT-l.

Mention detection performance. First, MD performance has improved significantly over the years. SpanBERT-l achieves an F-score of 88.2 in MD on OntoNotes, which is significantly higher than the best MD F-score achieved in the CoNLL-2012 shared task (77.7). Note that Stanford's mention detector performs substantially worse than those of the other resolvers, especially on ACE and KBP. The reason is that Stanford employs a rule-based MD system that was initially developed when the Stanford NLP Group participated in the CoNLL-2011 shared task, whereas in the other resolvers MD is jointly trained with CR. Overall, MD performance appears to have a significant impact on CR performance. In particular, joint MD and CR in the span-based resolvers seems to be a driving force behind the rapid coreference performance improvements we have seen in recent years.

[7] Owing to space limitations, we show only the most important scores in Table 3. The detailed results (e.g., B^3 and $CEAF_e$ results) can be found in the Appendix.

4 Using Oracles

Can the performance of coreference resolvers be further improved if we improve MD? Being able to answer this kind of questions is important: if further improvements in MD can result in significant gains in coreference performance, then future research efforts should perhaps be focused on MD.

To answer this kind of questions, we perform oracle experiments. Specifically, we provide a resolver with a particular type of perfect information (e.g., using gold mentions as input) and see how much performance improvement can be obtained.

4.1 Gold Mention Boundaries

Our first oracle experiment concerns training and testing our resolvers on gold mention boundaries. While this experiment has been conducted over the years by numerous researchers (e.g., Peng et al. (2015), Zhang et al. (2018)), we are primarily interested in understanding whether further improving an MD component that already has an F-score of more than 85% can improve coreference performance. For the four span-based models, we disable the component in the span representation layer that is responsible for proposing spans (i.e, mention boundaries) and instruct them to use gold mention spans instead. Note, however, that the representation of a span will be learned during training. In other words, although all resolvers are given gold mention spans, the span representations that will be used during resolution will still be different for different span-based resolvers.

Results, expressed in terms of the CoNLL score, are shown in the Gold Mention Boundaries col-

		Gold Mention Boundaries			Perfect Anaphoricity			Gold Entity Types		
	System	ACE	Onto.	KBP	ACE	Onto.	KBP	ACE	Onto.	KBP
1	Stanford	76.3	83.7	80.0	40.3	73.4	29.2	39.5	71.8	27.7
2	GloVe	79.3	86.1	81.6	71.5	76.2	69.8	69.6	71.4	67.9
3	ELMo	81.8	87.2	83.5	76.1	82.0	73.8	73.3	76.7	71.6
4	SpanBERT-b	84.9	90.5	85.7	79.3	84.4	76.3	77.4	79.3	73.8
5	SpanBERT-l	87.3	91.9	88.0	82.8	86.9	80.1	80.4	81.4	77.6

Table 4: CoNLL scores of the resolvers using gold mention boundaries, perfect anaphoricity, and gold entity types.

umn in Table 4. First, despite recent significant improvement in MD, these results suggest that coreference performance can still be significantly improved just by improving MD: for the best resolver (SpanBERT-l), the CoNLL score can be improved by 8.4–12.3% points. Second, the relative performances of the resolvers are consistent across the three datasets: the CoNLL scores increase as we go down the table. Since the four span-based resolvers use essentially the same (mention-ranking) model for resolution and the same algorithm for weight updates, their performance differences can be attributed largely to differences in the pretrained embeddings and the encoder. In addition, these results suggest that the coreference performance improvements we observed in recent years can be attributed to not only improved mention (boundary) detection but also improved resolution accuracy presumably as a result of better span representations.

4.2 Perfect Anaphoricity

Anaphoricity determination, a.k.a. discourse-new detection (Poesio et al., 2004), is the task of determining whether a mention is coreferent with another mention that appears earlier in the text. Being able to identify non-anaphoric mentions could improve the precision of coreference resolvers, as any antecedent chosen for them is erroneous.

In this oracle experiment, we provide a resolver with perfect anaphoricity information, meaning that we know for every entity mention whether it is anaphoric or not. We use this perfect anaphoricity information during resolution: we will resolve all and only those mentions that are anaphoric.

Results are shown in the Perfect Anaphoricity column of Table 4. A few points deserve mention. First, all resolvers improved on all datasets when provided with perfect anaphoricity information. These results imply that anaphoricity determination remains an important issue in CR research, and further improvements in anaphoricity can improve CR. However, the gains that state-of-the-art resolvers can achieve by improving anaphoricity determination are generally smaller than those by improving MD: the CoNLL scores of the span-based resolvers increase by 3.5–4.3% points on ACE, 4.4–9% points on OntoNotes, and 3.9–5% points on KBP. This is understandable, as MD is likely to improve both coreference precision and recall, whereas anaphoricity determination can only improve precision. Note that Stanford's poor performance on ACE and KBP is due to poor MD.

4.3 Gold Entity Types

In this experiment, we assume that a resolver is given gold entity types (i.e., semantic classes) such as PERSON, ORGANIZATION, and LOCATION. The set of entity types to be provided is corpus-dependent. As mentioned before, ACE and KBP only have seven and five entity types respectively. In OntoNotes, however, only named entities are annotated with (one of 18) entity types. Consequently, we automatically derive entity types for pronouns and nominals using gold coreference chains: if a pronoun or a nominal appears in a coreference cluster that contains a name, we derive its entity type from that of the name. This method allows us to derive the entity type of 36.4% of the nominals and 70% of the pronouns. Any pronoun or nominal whose entity type cannot be derived using this method will be assigned the entity type UNKNOWN. While this method does not provide full coverage, we will still be able to examine whether having access to perfect entity types on a subset of the mentions will enable us to improve the performance of a resolver on OntoNotes.

We use entity types during resolution. We disallow a candidate antecedent to be selected as the antecedent for a given anaphor if they have different entity types. Results are shown in the Gold Entity Types column in Table 4. As we can see, all resolvers improved on all datasets when provided with gold entity types. Compared with the gains achieved using gold mention spans or perfect anaphoricity, the gains that come with the use of gold entity types are smaller: for the span-based

resolvers, the CoNLL scores increase by 1.5–1.7% points on ACE, 1.8–3.7% points on OntoNotes, and 1.9–2.5% points on KBP. In other words, state-of-the-art resolvers can be improved by improving the determination of entity types.

These results are particularly interesting in light of a conundrum in entity CR: while some researchers have reported successes with improving entity CR using automatically computed semantic information (Ng, 2007), there have also been numerous failed attempts (Kehler et al., 2004; Durrett and Klein, 2013; Sapena et al., 2013). Although the semantic information we use in this paper is restricted to gold entity types, our results suggest that hand-annotated semantic information is indeed useful, and the (non-)utility of semantics for CR reported in earlier work could be attributed to the noise inherent in computing semantic information.

5 Results on Resolution Classes

To gain additional insights into the state-of-the-art resolvers, we analyze their performance on different types of entity mentions. More specifically, motivated by Stoyanov et al. (2009), we partition the *gold* mentions into different *resolution classes*. While previous work has focused mainly on three coarse-grained resolution classes (namely, pronouns, names, and nominal mentions), we employ the 13 fine-grained resolution classes defined by Rahman and Ng (2011b), as discussed below.

Names. Four classes are defined for gold names. (1) **e**: a name is assigned to this *exact string match* class if there is a preceding mention such that the two are coreferent and are the same string; (2) **p**: a name is assigned to this *partial string match* class if there is a preceding mention such that the two are coreferent and have some content words in common; (3) **n**: a name is assigned to this *no string match* class if there is no preceding mention such that the two are coreferent and have some content words in common; and (4) **na**: a name is assigned to this *non-anaphor* class if it is not coreferent with any preceding mention.

Nominal mentions. Four analogous resolution classes are defined for gold mentions whose head is a nominal: (5) **e**; (6) **p**; (7) **n**; and (8) **na**.

Pronouns. We have three pronoun classes. (9) **1/2**: 1st and 2nd person pronouns; (10) **G3**: gendered 3rd person pronouns (e.g., *she*); (11) **U3**: ungendered 3rd person pronouns; (12) **oa**: any anaphoric pronouns that do not belong to (9), (10), and (11) (e.g., relative pronouns); and (13) **na**: non-anaphoric pronouns (e.g., pleonastic pronouns).

Table 5 shows the performance of each resolver on each resolution class. To avoid overwhelming the reader, we only show the results of ELMo and SpanBERT-l, which will allow us to gain insights into what made SpanBERT-l better. Specifically, for each resolution class C, we show each resolver's MD recall (percentage of gold mentions in C that are correctly recalled) under **MD** and its resolution accuracy (percentage of correctly identified anaphors in C that are correctly resolved)[8] under **RA**. Under **Size** we show the percentage of gold mentions belonging to each resolution class.

First, if we consider only the three coarse-grained resolution classes, the results are perhaps not surprising: name resolution is the easiest and nominal resolution is the hardest.

Second, consider the 13 fine-grained resolution classes. By design, the names and the nominals in the 'e' class should be easier to resolve than those in 'p', which in turn should be easier to resolve than those in 'n'. The results are consistent with this intuition. Results on the anaphoric pronoun classes are also consistent with our intuition: 3rd person gendered pronouns are the easiest to resolve, followed by 1st/2nd person gendered pronouns and then ungendered 3rd person pronouns.

Third, these results reveal that the difficulty of anaphoricity determination stems primarily from pronouns: while resolution accuracies on non-anaphoric names and nominal mentions are above 89%, those on non-anaphoric pronouns are only between 65.9% and 77.6%. Note that we consider a non-anaphoric mention correctly "resolved" if it is resolved to the dummy antecedent.

Finally, SpanBERT-l has better resolution accuracies than ELMo for all resolution classes on all datasets. Encouragingly, the harder a resolution class is, the bigger the improvement is. These results clearly show that we are making progress on resolving anaphors that are traditionally considered difficult to resolve. Note that part of this improvement can be attributed to improved MD, which increases the likelihood that the correct antecedent of an anaphor is present in its list of candidate antecedents. Additional experiments are needed to determine the impact of improved MD on improvement in resolution accuracies, however.

[8] In other words, the resolution accuracy does not depend on anaphor recall and precision.

		ACE					OntoNotes					KBP				
		Size	ELMo		SpanB-l		Size	ELMo		SpanB-l		Size	ELMo		SpanB-l	
	Class	%	RA	MD	RA	MD	%	RA	MD	RA	MD	%	RA	MD	RA	MD
1	NAM-e	16.8	94.6	96.0	97.4	96.6	14.0	95.3	93.2	95.2	91.8	16.5	94.0	95.0	95.3	96.5
2	NAM-p	4.2	71.5	86.6	81.5	88.2	7.3	82.7	83.2	88.5	83.7	2.0	66.1	90.8	74.6	93.8
3	NAM-n	2.0	52.5	87.6	69.8	91.4	1.9	56.6	60.6	73.6	68.1	2.9	56.8	88.4	64.2	92.0
4	NAM-na	10.8	93.3	88.4	94.8	91.0	11.5	93.5	74.8	94.4	79.0	16.8	95.8	91.6	96.1	92.8
5	NOM-e	3.4	77.0	91.0	84.3	92.3	3.7	92.9	89.6	95.7	90.0	3.4	86.2	85.7	88.1	86.7
6	NOM-p	4.3	51.1	82.8	61.5	89.4	5.3	78.1	78.6	84.3	83.1	4.2	43.7	83.3	52.5	80.3
7	NOM-n	4.2	48.5	84.0	66.3	89.4	3.3	58.6	61.9	78.4	71.6	5.8	42.1	77.0	50.6	76.5
8	NOM-na	16.8	92.0	83.1	93.1	86.1	8.4	89.2	66.7	92.5	77.1	19.9	92.8	78.2	93.4	78.3
9	PRO-1/2	16.4	81.8	99.8	88.6	99.8	16.1	90.1	93.2	93.6	96.1	13.5	82.1	100	88.2	99.9
10	PRO-G3	5.7	88.5	99.8	94.1	100	10.7	91.6	99.6	95.9	99.3	6.5	88.3	100	94.7	99.7
11	PRO-U3	6.2	68.8	91.5	85.2	96.8	13.8	84.1	93.4	91.2	96.0	4.6	70.5	93.4	79.5	97.8
12	PRO-oa	3.9	52.9	80.3	70.3	86.1	2.2	57.0	57.3	69.1	70.0	1.2	68.3	69.8	72.0	72.7
13	PRO-na	5.3	73.9	87.3	77.6	90.6	1.8	65.9	80.5	69.6	84.3	2.8	66.5	93.5	72.6	92.7

Table 5: Results on resolution classes.

6 Sensitivity to Perturbed Inputs

Next, we conduct a series of experiments that involve perturbing the input. In each experiment, we (1) replace a certain kind of words/phrases in each training document with other words/phrases, (2) train a coreference model on these perturbed training documents, and (3) evaluate the output. Our goal is to gain insights into the behavior of state-of-the-art resolvers by examining how sensitive their performance is to perturbations in the input. Specifically, if performance drops significantly when a particular kind of words/phrases is replaced, that means the replaced words/phrases are important in the model learning process. Note that perturbations are only applied to the training documents; no changes are made to the test documents.

We divide the different kinds of perturbations into two broad categories, mention-internal perturbations and mention-external perturbations.

6.1 Mention-internal Perturbations

Mention-internal perturbations involve making changes to the words within an entity mention.

6.1.1 Perturbations to Names

We consider two kinds of perturbations to names.

Unseen names. We replace each name in a training document with a name that will highly unlikely appear in any test set. With this replacement, all the names in the test set will be unseen w.r.t. the training set. When trained on this perturbed training set, we can determine the extent to which the algorithms for learning coreference resolvers rely on memorizing seen names (as opposed to generalizing from their contexts) when performing MD and CR. Specifically, if a learner memorizes a lot, it will likely perform poorly on MD (i.e., its recall will suffer) and subsequently CR.

Perturbation Type	Example
Unseen names	Mr. Smith → Mr. Htims
Names of a different type	John Smith → New York
Unseen nominals	activist → tsivitca
Nominals of a different type	actor → plane
Nominals of the same type	wife → grandmother
Unseen verbs	support → troppus
Seen verbs	acquire → believe
Unseen adj/adv	directly → yltcerid
Seen adj/adv	organic → shredded

Table 6: Perturbation examples.

We perform name replacement in a deterministic manner: we replace each word in a name with another word in which the order of its characters is reversed. Note that person prefixes (e.g., "Mr."), organization words and suffixes ("Airlines", "Inc."), and location nouns (e,g, "River") will not be replaced, as the goal is to introduce unseen names rather than change the type of a name.[9] In addition, any word in a name that appears in a nominal mention in the training set will not be replaced. For instance, the word "Church" in "Baptist Church" appears in a nominal in the training set and therefore will not be replaced. This is done to ensure that only the "name" part of a mention will be changed to something that is not previously seen.[10]

Names of a different type. In this experiment, we replace each name, ne_1, in a training document with another name, ne_2, that satisfies two conditions. First, ne_2, like ne_1, should appear in the training set. This ensures that the number of names in the test set that will be unseen w.r.t. the train-

[9] These lists are available in the Appendix.

[10] Examples of this and other kinds of perturbations are shown in Table 6.

ing set will not change. Second, the two names should have different entity types. Importantly, the replacement is deterministic, meaning that (1) all occurrences of ne_1 will be replaced with the same name (i.e., ne_2), and (2) any name coreferent with ne_1 (but are not lexically identical to ne_1, such as "Trump" and "President Trump") will be replaced with a name coreferent with ne_2. These conditions together ensure that only the names and their types will change, but their coreference relationships will not. Note that the choice of ne_2 is random subject to these conditions. Due to the randomness involved in the selection of ne_2, we repeat the experiment three times and report the average result.

With this replacement, the resulting training documents may no longer make sense to a human reader, as a PERSON name may appear in a context for an ORGANIZATION name. In particular, the contexts in which a certain type of names (e.g., PERSON) appear in the training set will be different from those in which these names appear in the test set. This experiment will allow us to determine the extent to which a resolver makes use of contextual information when identifying coreference links involving names: if it makes heavy use of contextual information, we should see a considerable drop in resolver performance.

6.1.2 Perturbations to Nominal Mentions

We consider three kinds of perturbations to nominal mentions to determine the roles they play.

Unseen nominals. This experiment has the same setup as the "Unseen names" experiment above, except that we replace each nominal mention in the training set with another mention in which we reverse the order of the characters of each of its words. Note that this is a mention-internal perturbation, meaning that we replace all and only those nominals that are annotated as entity mentions, not all nominals in the training set.

Nominals of a different type. This experiment has the same setup as the "Names of a different type" experiment above, except that we replace nominal mentions rather than names. As in the previous experiment, we replace each nominal that is annotated as an entity mention in the training set.

Nominals of the same type. This experiment has the same setup as the "Nominals of a different type" experiment above, except that the nominal mention being replaced must have the same entity type as its replacement. This kind of perturbation is "milder" than the previous kind of perturbation, as a PERSON mention will continue to appear in a PERSON context after the replacement. In other words, if a machine learner does not pay attention to the semantic compatibility between a nominal mention and its context, then we should see little performance difference when a resolver is trained on this training set vs. the previous training set (i.e., the one from "Nominals of a different type").

6.2 Mention-external Perturbations

Mention-external perturbations involve making changes to the words outside a mention.

6.2.1 Perturbations to Verbs

We consider two kinds of perturbations to verbs to determine the role they play in resolution.

Unseen verbs. This experiment has the same setup as the two "Unseen" experiments above, except that we replace each verb in the training set that is not part of an entity mention.

Seen verbs. This experiment has the same setup as the "Names of a different type" experiment above, except that we replace verbs outside of entity mentions rather than names. In particular, the new verb is not constrained to have the same type as the verb being replaced: it can be any verb taken from the training set. Nevertheless, the replacement is deterministic: all occurrences of a given verb will be replaced with the same verb.

6.2.2 Perturbations to Adjectives & Adverbs

We consider two kinds of perturbations to adjectives and adverbs to determine the roles they play.

Unseen adjectives and adverbs. This experiment has the same setup as the three "Unseen" experiments above, except that we replace each adjective and adverb in the training set that is not part of an entity mention.

Seen adjectives and adverbs. This experiment has the same setup as the "Seen verbs" experiment above, except that we replace adjectives and adverbs outside of entity mentions rather than verbs.

6.3 Perturbation Results

Results of these experiments on the three datasets are shown in Table 7. As in Table 5, we only show the results of ELMo and SpanBERT-l. For each resolver, we show its CR CoNLL score and its MD F-score. To facilitate comparison, we show in row 1 the performance of the resolvers when the input is not perturbed.

		ACE				OntoNotes				KBP			
		ELMo		SpanB-l		ELMo		SpanB-l		ELMo		SpanB-l	
	Perturbation Type	CR	MD	CR	MD	CR	MD	CR	MD	CR	MD	CR	MD
1	No Perturbation	71.8	90.4	78.9	92.3	73.0	85.1	79.6	88.2	69.7	85.3	75.8	88.5
2	NAM-Unseen	68.4	88.6	74.9	90.8	46.2	63.3	65.7	77.2	62.1	79.7	65.7	79.7
3	NAM-DiffType	69.5	89.5	75.7	91.1	58.8	75.2	72.7	83.5	64.3	82.5	67.2	81.4
4	NOM-Unseen	67.6	86.5	73.8	88.7	57.4	73.6	70.7	81.9	60.6	76.4	67.8	81.7
5	NOM-DiffType	69.7	88.6	74.7	89.6	59.0	75.4	71.2	82.1	66.0	82.5	70.7	83.6
6	NOM-SameType	69.9	89.2	75.3	90.1	62.5	79.2	72.8	83.4	66.8	83.4	72.1	84.9
7	Verb-Unseen	70.6	89.4	76.6	91.1	64.7	79.7	75.3	86.2	68.2	84.2	74.0	86.9
8	Verb-Seen	71.6	90.1	77.1	91.1	67.2	81.1	77.0	87.0	68.8	84.7	74.4	87.1
9	Adj/Adv-Unseen	71.7	89.9	76.8	91.0	68.9	82.3	77.4	87.4	69.3	85.3	73.5	86.4
10	Adj/Adv-Seen	69.8	88.8	76.9	91.3	70.1	83.0	76.3	86.7	68.5	84.8	74.0	86.6

Table 7: Perturbation results.

A few points deserve mention. First, mention-internal perturbations (rows 2–6) triggered larger deterioration in CR performance than mention-external perturbations. These results suggest that the resolvers rely more on the mentions themselves than their contexts for resolution, which should not be surprising. Among the mention-internal perturbations, the biggest CR performance drops occur with the Unseen perturbations (rows 2 and 4), particularly those involving unseen names, followed by perturbations involving the replacement of a seen name or nominal with a different type. A closer inspection of the results reveals that there is a strong correlation between CR performance and MD performance: larger drops in CR performance are always accompanied by larger drops in MD performance. This sheds light on why the Unseen perturbations triggered the largest drop in CR performance: when all the names or nominal mentions in the test set are not seen in the training set, the mention detector is likely to perform poorly on the test set. In contrast, when they are replaced by mentions of a different entity type, the percentage of unseen mentions in the test set doesn't change, thus posing fewer problems for the mention detector.

As for the mention-external perturbations, no clear patterns emerged: while verb replacement (rows 7–8) has a greater impact than adjective and adverb replacement (rows 9–10) for OntoNotes, the same observation cannot be made for the other datasets. Moreover, while replacing a word with another seen word is generally expected to cause less harm to MD (and thus CR) performance than replacing a word with an unseen word, these experiments show that this is not necessarily the case. These results seem to suggest that the mention span learner is not particularly sensitive to the verbs, adjectives and adverbs that appear in the context.

Third, it is not easy to conclude which resolver is more robust to perturbations. While the drops in CR performance on ACE are fairly mild for both resolvers, we see bigger CR performance drops on the other two datasets. In particular, ELMo suffers from a bigger drop in performance than SpanBERT-l on OntoNotes, whereas the reverse is true on KBP.

Finally, while the two resolvers' MD performances are similar, SpanBERT-l's CR performance is always superior to ELMo's. These results reveal once again that the mention representations learned by SpanBERT-l are indeed better than those by ELMo as far as resolution is concerned.

7 Conclusions

While space limitations preclude a reiteration of all the observations we have made, we believe the key conclusions are: (1) the relative performances of the resolvers are consistent across datasets; (2) for each resolver, higher mention detection performance always yields better coreference performance; (3) the newest resolvers perform better because of not only improved mention detection, but also improved mention span representations, and they improved the resolution of both easy- and difficult-to-resolve anaphors; (4) all resolvers can be improved by improving mention detection, anaphoricity determination, and entity type detection; and (5) our perturbation results suggest that coreference performance is most sensitive to those words/phrases in the input that have the greatest impact on mention detection performance.

Acknowledgments

We thank the three anonymous reviewers for their comments. The paper's title was motivated by that of Stoyanov et al.'s ACL 2009 paper — the credit goes to them. This work was supported in part by NSF Grants IIS-1528037 and CCF-1848608.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the LREC Workshop on Linguistic Coreference*, pages 563–566.

Mohit Bansal and Dan Klein. 2012. Coreference semantics from Web features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 389–398.

Anders Björkelund and Jonas Kuhn. 2014. Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 47–57.

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

Dennis Connolly, John D. Burger, and David S. Day. 1994. A machine learning approach to anaphoric reference. In *Proceedings of International Conference on New Methods in Language Processing*, pages 255–261.

Pascal Denis and Jason Baldridge. 2008. Specialized models and ranking for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 660–669.

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

Eraldo Fernandes, Cícero dos Santos, and Ruy Milidiú. 2012. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 41–48.

Hannaneh Hajishirzi, Leila Zilles, Daniel S. Weld, and Luke Zettlemoyer. 2013. Joint coreference resolution and named-entity linking with multi-pass sieves. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 289–299.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60.

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

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5803–5808.

Ben Kantor and Amir Globerson. 2019. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 673–677.

Andrew Kehler, Douglas Appelt, Lara Taylor, and Aleksandr Simma. 2004. The (non)utility of predicate-argument frequencies for pronoun interpretation. In *Proceedings of the 2004 Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics Annual Meeting*, pages 289–296.

Jonathan K. Kummerfeld and Dan Klein. 2013. Error-driven analysis of challenges in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 265–277.

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.

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.

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

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

Nafise Sadat Moosavi and Michael Strube. 2016. 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.

Vincent Ng. 2007. Semantic class induction and coreference resolution. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 536–543.

Haoruo Peng, Kai-Wei Chang, and Dan Roth. 2015. A joint framework for coreference resolution and mention head detection. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 12–21.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Massimo Poesio, Olga Uryupina, Renata Vieira, Mijail Alexandrov-Kabadjov, and Rodrigo Goulart. 2004. Discourse-new detectors for definite description resolution: A survey and a preliminary proposal. In *Proceedings of the Conference on Reference Resolution and Its Applications*, pages 47–54.

Simone Paolo Ponzetto and Michael Strube. 2007. Knowledge derived from Wikipedia for computing semantic relatedness. *Journal of Artificial Intelligence Research*, 30:181–212.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–35.

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

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. 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.

Altaf Rahman and Vincent Ng. 2011a. Coreference resolution with world knowledge. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 814–824.

Altaf Rahman and Vincent Ng. 2011b. Narrowing the modeling gap: A cluster-ranking approach to coreference resolution. volume 40, pages 469–521.

Emili Sapena, Lluís Padró, and Jordi Turmo. 2013. A constraint-based hypergraph partitioning approach to coreference resolution. *Computational Linguistics*, 39(4):847–884.

Veselin Stoyanov, Nathan Gilbert, Claire Cardie, and Ellen Riloff. 2009. 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.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the Sixth Message Understanding Conference*, pages 45–52.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus LDC2006T06. Philadelphia: Linguistic Data Consortium.

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

A Lists of Prefixes and Suffixes

Table 8 shows the list of person prefixes, organization words and suffixes, and location nouns used in the perturbation experiments.

B Results from Different Evaluation Metrics

Recall that owing to space limitations, results of the different resolvers are only expressed in terms of the CoNLL score, the MUC F-score, the percentage of singleton clusters being recalled, and the mention detection F-score. Table 9 provides the detailed results on coreference expressed in terms of recall (R), precision (P) and F-score (F) that are via different evaluation metrics (i.e., MUC, B^3, $CEAF_e$, and LEA). In addition, mention detection performance is expressed in terms of R, P, and F.

As can be seen, regardless of which coreference evaluation metric is used, F-score consistently increases down the table for each dataset. These results provide suggestive evidence that the improvements achieved by each resolver over the previous ones are robust. As for mention detection, improvements in F-score are largely accompanied by improvements in both recall and precision.

Location	Person					Organization	
Mount	Acting	Czar	lt.	Commander	Reverend	laboratories	a.g.
Mt.	Adm	Democrat	Lt.	Commissioner	Reverends	laboratory	ag
River	Adm.	Deputy	maj	Commissioner	Revs	co.	a.b.
Bay	administrator	dr	Maj	commissioner	Revs	co	ab
Beach	admiral	Dr	maj.	Commodore	Revs.	cie	aktiebolag
Canal	Admiral	dr.	Maj.	congressman	Revs.	cie.	aktiengesellschaft
Cape	ambassador	Dr.	Major	Congressman	Sargent	cos.	n.v.
City	Ambassador	Drs	Marquis	Congressmen	secretary	corp	nv
County	Ambassadors	Drs.	Major	Congressperson	Secretary	corp.	bv
Desert	Archbishop	Ensign	mayor	Congresswoman	Secretary	inc.	b.v.
Gulf	Archbishops	Father	messrs	Congresswomen	sen	inc	p.c.
Harbor	Assistant	Fathers	messrs.	Ladies in waiting	Sen	ltd.	de c.v.
Inlet	Attorney	First Lady	Minister	Ladies-in-waiting	sen.	ltd	de cv
Island	Bishop	gen	mr	Lady in waiting	Sen.	ltda.	b.d.d.p.
Islands	Bishops	Gen	Mr	Lady-in-waiting	senator	ltda	bddp
Islet	Brig	gen.	mr.	Leader	Senator	l.p.	Airlines
Islets	Brig.	Gen.	Mr.	Leaders	Senators	lp	Airways
Mountain	brigadier	gov	mrs	lieutenant	sens	Associates	Brothers
Mountains	Brigadier	Gov	Mrs	Lieutenant	Sens	Assoc.	Developments
Ocean	Capt	gov.	mrs.	Mission Specialist	sens.	group	Partners
Park	Capt.	Gov.	Mrs.	Prime Minister	sergeant	groupe	Properties
Peninsula	Captain	Governor	ms	Prime minister	sgt	grupo	Stores
Plains	CEO	Governors	ms.	Princess	Sgt	bros	
Pond	CFO	Govs	Mssrs	prof	Sgt.	bros.	
Province	chairman	Holiness	Mssrs.	prof.	Sir	bancorp	
Road	Chancellor	Hon	Officer	Queen	stg.	bancorp.	
Roads	Chancellors	Hon.	officer	Queens	Undersecretaries	sdn	
Sea	Chief	Honorable	officers	rep	Undersecretary	sdn.	
Shore	Cmdr	Honorable	Petty	Rep	Vicar	bhd	
Straits	col	Inspector	Premier	rep.	Vicars	bnd.	
Town	col.	Jr	Premiers	Rep.	representative	plc	
Valley	colonel	Jr.	Pres	Repr	Representative	plc.s.a.	
	Comdr	Judge	Pres.	Repr.	Representatives	sa	
	Comdr.	Judges	Prime	president	reps	M.e.T.A.	
	Consul	Junior	Rev	President	Reps	g.m.b.h.	
	COO	King	Rev.	president	reps.	gmbh	
	Corporal	Kings	Lord	Presidents	Republican	s.p.a.	
	Crpl	Crpl.	Crprl	lt	Lt	c.a.	

Table 8: Lists of prefixes and suffixes.

	Coreference													MD		
	MUC			B^3			$CEAF_e$			CoNLL	LEA					
	R	P	F	R	P	F	R	P	F	F	R	P	F	R	P	F
	ACE															
Stanford	45.0	81.2	57.9	36.6	34.8	35.7	57.0	13.9	22.4	38.6	29.5	25.1	27.1	69.3	39.9	50.6
GloVe	71.8	82.1	76.6	63.4	71.7	67.3	69.1	53.3	60.2	68.0	55.8	60.1	57.9	88.2	86.6	87.4
ELMo	76.1	83.3	79.5	67.2	74.2	70.5	71.9	60.1	65.5	71.8	59.8	63.7	61.7	90.9	89.9	90.4
SpanBERT-b	81.9	84.2	83.1	74.1	74.7	74.4	73.5	66.3	69.7	75.7	67.7	66.7	67.2	92.6	90.6	91.6
SpanBERT-l	84.0	86.9	85.4	77.5	79.6	78.5	76.6	69.2	72.7	78.9	71.9	72.3	72.1	93.1	91.5	92.3
	OntoNotes															
Stanford	70.2	79.0	74.3	57.7	69.8	63.2	55.1	63.6	59.1	65.5	54.0	66.0	59.4	75.4	85.4	80.1
GloVe	72.7	81.3	76.7	60.3	72.8	66.0	57.9	65.8	61.6	68.1	57.1	69.5	62.7	77.5	87.0	82.0
ELMo	79.5	81.4	80.5	69.4	72.2	70.8	67.2	68.2	67.7	73.0	66.4	69.1	67.7	84.2	86.0	85.1
SpanBERT-b	83.1	84.3	83.7	75.3	76.2	75.8	71.2	74.6	72.9	77.4	72.8	73.8	73.3	86.2	88.1	87.1
SpanBERT-l	84.8	85.8	85.3	77.9	78.3	78.1	74.2	76.4	75.3	79.6	75.7	76.2	75.9	87.6	88.9	88.2
	KBP															
Stanford	43.3	56.0	48.8	38.8	13.2	19.7	48.7	5.7	10.2	26.2	29.6	9.4	14.3	64.1	16.6	26.4
GloVe	70.5	72.1	71.3	66.1	66.2	66.1	69.4	53.2	60.3	65.9	57.2	54.6	55.9	86.9	78.3	82.4
ELMo	74.7	72.9	73.8	70.6	68.5	69.5	73.1	60.0	65.9	69.7	62.0	58.3	60.1	89.7	81.2	85.3
SpanBERT-b	78.1	73.3	75.6	74.5	67.7	70.9	73.4	62.4	67.5	71.3	66.0	57.9	61.6	91.1	81.9	86.3
SpanBERT-l	79.5	80.7	80.1	75.9	76.3	76.1	75.8	67.4	71.3	75.8	68.8	67.1	67.9	90.5	86.7	88.5

Table 9: Results of the resolvers according to different evaluation metrics on the three coreference datasets.