Error-Driven Analysis of Challenges in Coreference Resolution

Jonathan K. Kummerfeld and **Dan Klein**
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720, USA
{jkk,klein}@cs.berkeley.edu

Abstract

Coreference resolution metrics quantify errors but do not analyze them. Here, we consider an automated method of categorizing errors in the output of a coreference system into intuitive underlying error types. Using this tool, we first compare the error distributions across a large set of systems, then analyze common errors across the top ten systems, empirically characterizing the major unsolved challenges of the coreference resolution task.

1 Introduction

Metrics produce measurements that concisely summarize performance on the full range of error types, and for coreference resolution there has been extensive work on developing effective metrics (Luo, 2005; Recasens and Hovy, 2011). However, it is also valuable to tease apart the errors to understand their relative importance.

Previous investigations of coreference errors have focused on quantifying the importance of subtasks such as named entity recognition and anaphoricity detection, typically by measuring accuracy improvements when partial gold annotations are provided (Stoyanov et al., 2009; Pradhan et al., 2011; Pradhan et al., 2012). For coreference resolution the drawback of this approach is that decisions are often interdependent, and so even partial gold information is extremely informative. Also, previous work only considered errors by counting links, which does not capture certain errors in a natural way, e.g. when a system incorrectly divides a large entity into two parts, each with multiple mentions. Recent work has considered some of these issues, but only with small scale manual analysis (Holen, 2013).

We present a new tool that automatically classifies errors in the standard output of any coreference resolution system. Our approach is to identify changes that convert the system output into the gold annotations, and map the steps in the conversion onto linguistically intuitive error types. Since our tool uses only system output, we are able to classify errors made by systems of any architecture, including both systems that use link-based inference and systems that use global inference methods.

Using our tool we perform two studies to understand similarities and differences between systems. First, we compare the error distributions on coreference resolution of all of the systems from the CoNLL 2011 shared task plus several publicly available systems. This comparison adds to the analysis from the shared task by illustrating the substantial variation in the types of errors different systems make. Second, we investigate the aggregate behavior of ten state-of-the-art systems, providing a detailed characterization of each error type. This investigation identifies key outstanding challenges and presents the impact that solving each of them would have in terms of changes in the standard coreference resolution metrics.

We find that the best systems are not best across all error types, that a large proportion of span errors are due to superficial parse differences, and that the biggest performance loss is on missed entities that contain a small number of mentions.

This work presents a comprehensive investigation of common errors in coreference resolution, identifying particular issues worth focusing on in future research. Our analysis tool is available at code.google.com/p/berkeley-coreference-analyser/.

2 Background

Most coreference work focuses on accuracy improvements, as measured by metrics such as MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), CEAF (Luo, 2005), and BLANC (Recasens and Hovy, 2011). The only common forms of further analysis are results for anaphoricity detection and scores for each mention type (nominal, pronoun, proper). Two exceptions are: the detailed analysis of the Reconcile system by Stoyanov et al. (2009), and the multi-system comparisons in the CoNLL shared task reports (Pradhan et al., 2011, 2012).

A common approach to performance analysis is to calculate scores for nominals, pronouns and proper names separately, but this is a very coarse division (Ng and Cardie, 2002; Haghighi and Klein, 2009). More fine consideration of some subtasks does occur, for example, anaphoricity detection, which has been recognized as a key challenge in coreference resolution for decades and regularly has separate results reported (Paice and Husk, 1987; Sobha et al., 2011; Yuan et al., 2012; Björkelund and Farkas, 2012; Zhekova et al., 2012). Some work has also included anecdotal discussion of specific error types or manual classification of a small set of errors, but these approaches do not effectively quantify the relative impact of different errors (Chen and Ng, 2012; Martschat et al., 2012; Haghighi and Klein, 2009). In a recent paper, Holen (2013) presented a detailed manual analysis that considered a more comprehensive set of error types, but their focus was on exploring the shortcomings of current metrics, rather than understanding the behavior of current systems.

The detailed investigation presented by Stoyanov et al. (2009) is the closest to the work we present here. First, they measured accuracy improvements when their system was given gold annotations for three subtasks of coreference resolution: mention detection, named entity recognition, and anaphoricity detection. To isolate other types of errors they defined resolution classes, based on both the type of a mention, and properties of possible antecedents (for example, nominals that have a possible antecedent that is an exact string match). For each resolution class they measured performance while giving the system gold annotations for all other classes. While this approach is effective at characterizing variations between the nine classes they defined, it misses the cascade effect of errors that only occur when all mentions are being resolved at once.

President Clinton$_1$ is questioning the legitimacy of George W. Bush's election victory. Speaking last night to Democratic supporters in Chicago, **he** said Bush won the election only because Republicans stopped the vote-counting in Florida, and Mr. Clinton$_1$ praised Al Gore's campaign manager, Bill Daley, for the way he handled the election. "I$_2$ want to thank Bill Daley for his exemplary service as Secretary of Commerce. He was brilliant. I$_2$ think he did a brilliant job in leading Vice President Gore to victory myself$_2$."

Figure 1: Two coreference errors. Mentions are underlined and subscripts indicate entities. One error is a mention missing from the system output, *he*. The other is the division of references to Bill Clinton into two entities.

The only multi-system comparisons are the CoNLL task reports (Pradhan et al., 2011, 2012), which explored the impact of mention detection and anaphoricity detection through subtasks with different types of gold annotation. With a large set of systems, and well controlled experimental conditions, the tasks provided a great snapshot of progress in the field, which we aim to supplement by characterizing the major outstanding sources of error.

This work adds to previous investigations by providing a comprehensive and detailed analysis of errors. Our tool can automatically analyze any system's output, giving a reliable estimate of the relative importance of different error types.

3 Error Classification

When inspecting the output of coreference resolution systems, several types of errors become immediately apparent: entities that have been divided into pieces, spurious entities, non-referential pronouns that have been assigned antecedents, and so on. Our goal in this work is to automatically assign intuitive labels like these to errors in system output.

A simple approach, refining results by measuring the accuracy of subsets of the mentions, can be misleading. For example, in Figure 1, we can intuitively see two pronoun related mistakes: a missing mention (*he*), and a divided entity where the two pieces are the blue pronouns (I_2, I_2, $myself_2$) and the red proper names (*President Clinton*$_1$, *Mr. Clinton*$_1$).

Simply counting the number of incorrect pronoun links would miss the distinction between the two types of mistakes present.

One question in designing an error analysis tool like ours is whether to operate on just system output, or to also consider intermediate system decisions. We focused on using system output because other methods cannot uniformly apply to the full range of coreference resolution decoding methods, from link based methods to global inference methods.

Our overall approach is to transform the system output into the gold annotations, then map the changes made in the conversion process to errors. The transformation process is presented in Section 3.1 and Figure 2, and the mapping process is described in Section 3.2 and Figure 3.

3.1 Transformations

The first part of our error classification process determines the changes needed to transform the system output into the gold annotations. This five stage process is described below, and an abstract example is presented in Figure 2.

1. **Alter Span** transforms an incorrect system mention into a gold mention that has the same head token. In Figure 2 this stage is demonstrated by a mention in the leftmost entity, which has its span altered, indicated by the change from an X to a light blue circle.
2. **Split** breaks the system entities into pieces, each containing mentions from a single gold entity. In Figure 2 there are three changes in this stage: the leftmost entity is split into a red piece and a light blue piece, the middle entity is split into a dark red piece and an X, and the rightmost entity is split into singletons.
3. **Remove** deletes every mention that is not present in the gold annotations. In Figure 2 this means the four singleton X's are removed.
4. **Introduce** creates a singleton entity for each mention that is missing from the system output. In Figure 2 this stage involves the introduction of a light blue mention and two white mentions.
5. **Merge** combines entities to form the final, completely correct, set of entities. In Figure 2 the two red entities are merged, the singleton blue entity is merged with the rest of the blue entity, and the two white mentions are merged.

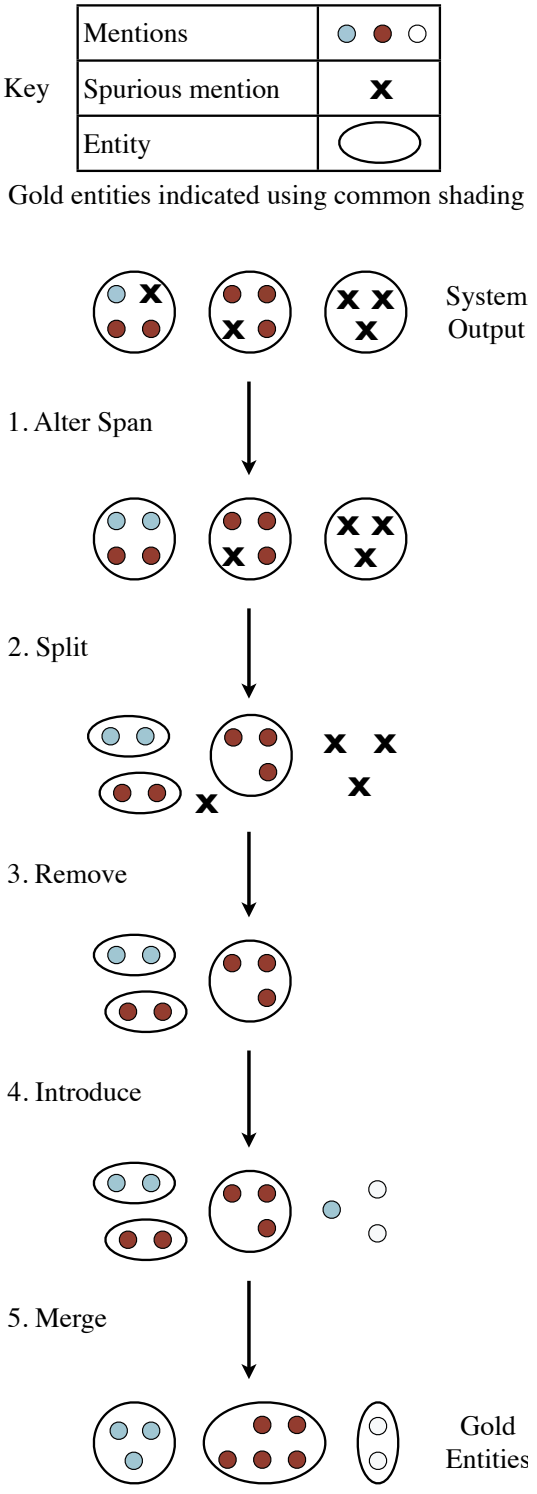

Figure 2: Abstract example of the transformation process that converts system output (at the top) to gold annotations (at the bottom).

	Operation(s)	Error	System	Gold
i)	Alter Span	Span error	*Gorbachev*	*Soviet leader Gorbachev*
ii)	Multiple Introduces and Merges	Missing Entity	- -	*the pills* *the tranquilizing pills*
iii)	Multiple Splits and Removes	Extra Entity	*human rights* *Human Rights*	- -
iv)	Introduce and Merge	Missing Mention	*the Arab region* *the region* -	*the Arab region* *the region* *it*
v)	Split and Remove	Extra Mention	*her story* *this* *it*	*her story* *this* -
vi)	Merge	Divided Entity	$Iraq_1$ $this\ nation_2$ $the\ nation_2$ its_1	$Iraq_1$ $this\ nation_1$ $the\ nation_1$ its_1
vii)	Split	Conflated Entities	$Mohammed\ Rashid_1$ $the\ Rashid\ case_1$ $Rashid_1$ $the\ case_1$	$Mohammed\ Rashid_1$ $the\ Rashid\ case_2$ $Rashid_1$ $the\ case_2$

Figure 3: Examples of the error types. In examples (i) - (iv) and (vi) the system output contains a single entity. When multiple entities are involved, they are marked with subscripts. Mentions are in the order in which they appear in the text. All examples are from system output on the dev set of the CoNLL task.

One subtle point in the split stage is how to record an entity being split into several pieces. This could either be a single operation, one entity being split into N pieces, or $N-1$ operations, each involving a single piece being split off from the rest of the entity. We use the second approach, as it fits more naturally with the error mapping we describe in the following section. Similarly, for the merge operation, we record N entities being merged as $N-1$ operations.

3.2 Mapping

The operations in Section 3.1 are mapped onto seven error types. In some cases, a single change maps onto a single error, while in others a single error represents several closely related operations from adjacent stages in the error correction process. The mapping is described below and in Figure 3.

1. **Span Error**. Each Alter Span operation is mapped to a Span Error, e.g. in Figure 3(i), the system mention *Gorbachev* is replaced by the annotated mention *Soviet leader Gorbachev*.
2. **Missing Entity**. A set of Introduce and Merge operations that forms an entirely new entity, e.g. the white entity in Figure 2, and *the pills* in Figure 3(ii). This error is still assigned if the new entity includes pronouns that were already present in the system output. The reasoning for this is that most pronouns in the corpus are coreferent, so including just the pronouns from an entity is not meaningfully different from missing the entity entirely.
3. **Extra Entity**. A set of Split and Remove operations that completely remove an entity, e.g. the rightmost entity in Figure 2, and Figure 3(iii). As for the Missing Entity error type, this error is still assigned if the original entity contained pronouns that were valid.
4. **Missing Mention**. An Introduce and a Merge that apply to the same mention, e.g. *it* in Figure 3(iv), and the blue mention in Figure 2.
5. **Extra Mention**. A Split and a Remove that apply to the same mention, e.g. *it* in Figure 3(v), and the X in the red entity in Figure 2.
6. **Divided Entity**. Each remaining Merge operation is mapped to a Divided Entity error, e.g. Figure 3(vi), and the red entity in Figure 2.
7. **Conflated Entities**. Each remaining Split operation is mapped to a Conflated Entity error, e.g. Figure 3(vii), and the blue and red entities in Figure 2.

4 Methodology

Our tool processes the CoNLL task output, with no other information required. During development, and when choosing examples for this paper, we used the development set of the CoNLL shared task (Hovy et al., 2006; Pradhan et al., 2007; Pradhan et al., 2011). The results we present in the rest of the paper are all for the test set. Using the development set would have been misleading, as the entrants in the shared task used it to tune their systems.

4.1 Systems

We analyzed all of the 2011 CoNLL task systems, as well as several publicly available systems. For the shared task systems we used the output data from the task itself, provided by the organizers. For the publicly available systems we used the default configurations. Finally, we included another run of the Stanford system, with their OntoNotes-tuned parameters (STANFORD-T).

The publicly available systems we used are: BERKELEY (Durrett and Klein, 2013), IMS (Björkelund and Farkas, 2012), STANFORD (Lee et al., 2013), RECONCILE (Stoyanov et al., 2010), BART (Versley et al., 2008), UIUC (Bengtson and Roth, 2008), and CHERRYPICKER (Rahman and Ng, 2009). The systems from the shared task are listed in Table 1 and in the references.

5 Broad System Comparison

Table 1 presents the frequency of errors for each system and F-Scores for standard metrics[1] on the test set of the 2011 CoNLL shared task. Each bar is filled in proportion to the number of errors the system made, with a full bar corresponding to the number of errors listed in the bottom row.

The metrics provide an effective overall ranking, as the systems with high scores generally make fewer errors. However, the metrics do not convey the significant variation in the types of errors systems make. For example, YANG and CHARTON are assigned almost the same scores, but YANG makes more than twice as many Extra Mention errors.

The most frequent error across all systems is Divided Entity. Unlike parsing errors (Kummerfeld et al., 2012), improvements are not monotonic, with better systems often making more errors of one type when decreasing the frequency of another type.

One outlier is the Irwin et al. (2011) system, which makes very few mistakes in five categories, but many in the last two. This reflects a high precision, low recall approach, where clusters are only formed when there is high confidence.

The third section of Table 1 shows results for systems that were run with gold noun phrase span information. This reduces all errors slightly, though most noticeably Extra Mention, Missing Mention, and Span Error. On inspection of the remaining Span Errors we found that many are due to inconsistencies regarding the inclusion of the possessive.

The final section of the table shows results for systems that were provided with the set of mentions that are coreferent. In this setting, three of the error types are not present, but there are still Missing Mentions and Missing Entities because systems do not always choose an antecedent, leaving a mention as a singleton, which is then ignored.

While this broad comparison gives a complete view of the range of errors present, it is still a coarse representation. In the next section, we characterize the common errors on a finer level by breaking down each error type by a range of properties.

6 Common Errors

To investigate the aggregate state of the art, in this section we consider results averaged over the top ten systems: CAI, CHANG, IMS, NUGUES, SANTOS, SAPENA, SONG, STANFORD-T, STOYANOV, URYUPINA-OPEN.[2] These systems represent a broad range of approaches, all of which are effective.

In each section below, we focus on one or two error types, characterizing the mistakes by a range of properties. We then consider a few questions that apply across multiple error types.

6.1 Span Errors

To characterize the Span Errors, we considered the text that is in the gold mention, but not the system

[1] CEAF and BLANC are not included as the most recent version of the CoNLL scorer (v5) is incorrect, and there are no standard implementations available.

[2] For systems that occur multiple times in Table 1, we only use the best instance. The BERKELEY system was not included as it had not been published at submission time.

System	Metric F-Scores Mention	MUC	B^3	Span Error	Conflated Entities	Extra Mention	Extra Entity	Divided Entity	Missing Mention	Missing Entity
PUBLICLY AVAILABLE SYSTEMS										
BERKELEY	75.57	66.43	66.17							
IMS	72.96	64.71	64.73							
STANFORD-T	71.21	61.40	63.06							
STANFORD	58.56	48.37	56.42							
RECONCILE	46.45	49.40	54.90							
BART	56.61	46.00	52.56							
UIUC	50.60	45.21	52.88							
CHERRYPICKER	41.10	40.71	51.39							
CONLL, PREDICTED MENTIONS										
LEE-OPEN	70.94	61.03	62.96							
LEE	70.70	59.56	61.88							
SAPENA	43.20	59.54	61.28							
SONG	67.26	59.95	60.08							
CHANG	64.86	57.13	61.75							
CAI-OPEN	67.45	57.86	60.89							
NUGUES	68.96	58.61	59.75							
URYUPINA-OPEN	68.39	57.63	58.74							
SANTOS	65.45	56.65	59.48							
STOYANOV	67.78	58.43	57.35							
HAO	64.30	54.46	55.82							
YANG	63.93	52.31	55.85							
CHARTON	64.36	52.49	55.61							
KLENNER-OPEN	62.28	49.86	55.62							
SOBHA	64.83	50.48	54.85							
ZHOU	62.31	48.96	53.42							
KOBDANI	61.03	48.62	53.00							
ZHANG	61.13	47.88	52.76							
XINXIN	61.92	46.62	51.50							
KUMMERFELD	62.72	42.70	50.05							
IRWIN-OPEN	35.27	27.21	44.29							
ZHEKOVA	48.29	24.08	41.42							
IRWIN	26.67	19.98	42.73							
CONLL, GOLD NP SPANS										
LEE-OPEN	75.39	65.39	65.88							
LEE	75.16	63.90	64.70							
NUGUES	72.42	62.12	61.67							
CHANG	67.91	59.77	62.97							
SANTOS	67.80	59.52	61.35							
STOYANOV	70.29	61.53	59.07							
SONG	66.68	55.48	58.04							
KOBDANI	66.08	53.94	55.82							
ZHANG	64.89	51.64	54.77							
ZHEKOVA	62.67	35.22	45.80							
CONLL, GOLD MENTIONS										
LEE-OPEN	90.93	81.56	75.95							
CHANG	99.97	82.52	73.68							
Most Errors				2410	3849	2744	5290	4789	2026	3237

Table 1: Counts for each error type on the test set of the 2011 CoNLL task. Bars indicate the number of errors, with white as zero and fully filled as the number in the *Most Errors* row. -OPEN indicates a system using external resources.

Type	Missing	Extra
NP	65.8	45.0
POS	12.4	96.9
,	71.2	22.4
SBAR	55.9	1.9
PP	46.2	10.3
DT	17.0	35.9
Total	271.1	224.6

Table 2: Counts of Span Errors grouped by the label over the extra/missing part of the mention.

mention (missing text), and vice versa (extra text). We then found nodes in the gold parse that covered just this extra/missing text, e.g. in Figure 3(i) we would consider the node over *Soviet leader*. In Table 2 we show the most frequent parse nodes.

Some of these differences are superficial, such as the possessive and the punctuation. Others, such as the missing PP and SBAR cases, may be due to parse errors. Of the system mentions involved in span errors, 27.0% do not correspond to a node in the gold parse. The frequency of punctuation errors could also be parse related, because punctuation is not considered in the standard parser evaluation.

Overall it seems that span errors can best be dealt with by improving parsing, though it is not possible to completely eliminate these errors because of inconsistent annotations.

6.2 Extra Mention and Missing Mention

We consider Extra and Missing Mentions together as they mirror each other, forming a precision-recall tradeoff, where a high precision system will have fewer Extra Mentions and more Missing Mentions, and a high recall system will have the opposite.

Table 3 divides these errors by the type of mention involved and presents some of the most frequent Extra Mentions and Missing Mentions. For the corpus statistics we count as mentions all NP spans in the gold parse plus any word tagged with PRP, WP, WDT, or WRB (following the definition of gold mention boundaries for the CoNLL tasks).

The mentions *it* and *you* are the most common errors, matching observations from several of the papers cited in Section 2. However, there is a surprising imbalance between Extra and Missing cases, e.g. *it* accounts for a third of the extra errors, but only 12% of the Missing errors. This imbalance may

	Av. Errors		Corpus Stats	
Mention	Extra	Missing	Count	% Coref.
Proper Name	281.6	297.7	6915	59.0
Nominal	484.2	516.5	33328	15.9
Pronoun	390.7	323.3	9926	69.7
it	130.4	38.9	1211	57.1
you	85.2	55.9	1028	44.9
we	39.6	19.6	691	64.7
us	23.2	3.2	242	23.6
that	13.8	13.4	2010	11.5
they	9.6	39.5	738	94.3
their	8.6	21.5	410	95.1
Total	1156.5	1137.5	50169	32.5

Table 3: Counts of Missing and Extra Mention errors by mention type, and the most common mentions.

	Proper Name		Nominal	
	Extra	Missing	Extra	Missing
Text match	145.2	163.6	171.2	96.1
Head match	56.8	70.7	149.6	166.0
Other	79.6	63.4	163.4	254.4
NER Matches	143.4	174.4	23.0	32.0
NER Differs	6.6	6.1	2.4	0.0
NER Unknown	131.6	117.2	458.8	484.5
Total	281.6	297.7	484.2	516.5

Table 4: Counts of Extra and Missing Mentions, grouped by properties of the mention and the entity it is in.

be the result of systems being tuned to the metrics, which seem to penalize Missing Mentions more than Extra Mentions (shown in Section 6.7).

In Table 4 we consider the Extra Mention errors and Missing Mention errors involving proper names and nominals. The top section counts errors in which the mention involved in the error has an exact string match with a mention in the cluster, or whether it has just a head match. The second section of the table considers the named entity annotations in OntoNotes, counting how often the mention's type matches the type of the cluster.

In all cases shown in the table it appears that systems are striking a balance between these two types of errors. One exception may be the use of exact string matching for nominals, which seems to be biased towards Extra Mentions.

For these two error types, our observations agree with previous work: the most common specific error is the identification of pleonastic pronouns, named entity types are of limited use, and head matching is already being used about as effectively as it can be.

Composition			Av. Errors	
Name	Nom	Pro	Extra	Missing
0	1	1	70.7	271.6
1	0	1	13.2	28.1
1	1	0	26.6	86.2
2	0	0	61.3	89.3
0	2	0	512.0	347.9
0	0	2	110.9	13.6
3+	0	0	14.7	14.4
0	3+	0	154.8	65.9
0	0	3+	91.0	18.1
Other			51.8	216.4
Total			1107.0	1151.5

Table 5: Counts of Extra and Missing Entity errors, grouped by the composition of the entity (Names, Nominals, Pronouns).

Match	Type	Extra	Missing
Exact	Proper Name	51.4	42.2
	Nominal	338.3	49.5
	Pronoun	141.9	10.3
Head	Proper Name	14.4	27.3
	Nominal	234.7	129.0
None	Proper Name	10.2	34.2
	Nominal	92.8	235.3
	Pronoun	60.0	21.4

Table 6: Counts of Extra and Missing Entity errors grouped by properties of the mentions in the entity.

6.3 Extra Entities and Missing Entities

In this section, we consider the errors that involve an entire entity that was either missing from the system output or does not exist in the annotations.

Table 5 counts these errors based on the composition of the entity. There are several noticeable differences between the two error types, e.g. for entities containing one nominal and one pronoun (row 0 1 1) there are far more Missing errors than Extra errors, while entities containing two pronouns (row 0 0 2) have the opposite trend.

It is clear that entities consisting of a single type of mention are the primary source of these errors, accounting for 85.3% of the Extra Entity errors, and 47.7% of Missing Entity errors. Table 6 shows counts for these cases divided into three groups: when all mentions are identical, when all mentions have the same head, and the rest.

Nominals are the most frequent type in Table 6, and have the greatest variation across the three sections of the table. For the Extra column, Exact match cases are a major challenge, accounting for over half of the nominal errors. These errors include cases like the example below, where two mentions are not considered coreferent because they are generic:

everybody tends to mistake the part for the whole.
Here, mistaking the part for the whole is ...

For missing entities we see the opposite trend, with Exact match cases accounting for less than 12% of nominal errors. Instead, cases with no match are the greatest challenge, such as this example, which requires semantic knowledge to correctly resolve:

The charges related to her sale of ImClone stock.
She sold the share a day before ...

The other common case in Table 5 is an entity containing a pronoun and a nominal. In Table 7 we present the most frequent pronouns for this case and the similar case involving a pronoun and a name.

Mention	Extra	Missing
that	6.9	99.7
it	47.7	47.8
this	0.9	36.2
they	3.8	29.1
their	2.1	23.5
them	0.9	13.8
Any pronoun	83.9	299.7

Table 7: 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.

One way of interpreting these errors is from the perspective of the pronoun, which is either incorrectly coreferent (Extra), or incorrectly non-coreferent (Missing). From this perspective, these errors are similar in nature to those described by Table 3. However, the distribution of errors is quite different, with *it* being balanced here where previously it skewed heavily towards extra mentions, while *that* was balanced in Table 3 but is skewed towards being part of Missing Entities here.

Extra Entity errors and Missing Entity errors are particularly challenging because they are dominated by entities that are either just nominals, or a nominal and a pronoun, and for these cases the string matching features are often misleading. This implies that reducing Extra Entity and Missing Entity errors will require the use of discourse, context, and semantics.

Incorrect Part			Rest of Entity			Av. Errors	
Na	No	Pr	Na	No	Pr	Conflated	Divided
-	-	1+	-	-	1+	312.7	69.9
-	-	1+	-	1+	1+	238.5	179.8
-	-	1+	-	1+	-	189.6	549.3
-	1+	-	-	1+	-	181.5	156.5
-	-	1+	1+	1+	1+	143.6	181.5
-	-	1+	1+	-	1+	109.7	150.5
-	-	1+	1+	-	-	60.0	136.5
Other						454.8	657.7
Total						1690.4	2081.7

Table 8: Counts of Conflated and Divided entities errors grouped by the Name / Nominal / Pronoun composition of the parts involved.

6.4 Conflated Entities and Divided Entities

Table 8 breaks down the Conflated Entities errors and Divided Entity errors by the composition of the part being split/merged and the rest of the entity involved. Each 1+ indicates that at least one mention of that type is present (Name / Nominal / Pronoun).

Clearly pronouns being placed incorrectly is the biggest issue here, with almost all of the common errors involving a part with just pronouns. It is also clear that not having proper names in the rest of the entity presents a challenge. One particularly noticeable issue involves entities composed entirely of pronouns, which are often created by systems conflating the pronouns of two entities together.

Table 8 aggregates errors by the presence of different types of mentions. Aggregating instead by the exact composition of the incorrect part being conflated or divided we found that instances with a part containing a single pronoun account for 38.9% of conflated cases and 35.8% of divided cases.

Finally, it is worth noting that in many cases a part is both conflated with the wrong entity, and divided from its true entity. Only 12.6% of Conflated Entity errors led to a complete gold entity with no other errors, and only 21.3% of Divided Entity errors came from parts that were not involved in another error.

Conflated Entities and Divided Entities are dominated by pronoun link errors: cases where a pronoun was placed in the wrong entity. Finding finer characterizations of these errors is difficult, as almost any division produces sparse counts, reflecting the long tail of mistakes that make up these two error types.

Gold	System Decision	Count
Cataphoric	Same referent	10.6
	Different referent	13.4
	Not cataphoric	208.2
	Not present	42.8
Not cataphoric	Cataphoric	46.2
Not present	Cataphoric	186.8

Table 9: Occurrence of mistakes involving cataphora.

6.5 Cataphora

Cataphora (when an anaphor precedes its antecedent) is a pronoun-specific problem that does not fit easily in the common left-to-right coreference resolution approach. In the CoNLL test set, 2.8% of the pronouns are cataphoric. In Table 9 we show how well systems handle this challenge by counting mentions based on whether they are cataphoric in the annotations, are cataphoric in the system output, and whether the antecedents match.

Systems handle cataphora poorly, missing almost all of the true instances, and introducing a large number of extra cases. However, this issue is a fairly small part of the task, with limited metric impact.

6.6 Entity Properties

Gender, number, person, and named entity type are properties commonly used in coreference resolution systems. In some cases, two mentions with different properties are placed in the same entity. Some of these cases are correct, such as variation in person between mentions inside and outside of quotes. However, many of these cases are errors. In Table 11 we present the percentage of entities that contain mentions with properties of more than one type. For named entity types we considered the annotations in OntoNotes; for the other properties we derive them from the pronouns in each cluster.

For all of the properties, there are many entities that we could not assign a value to, either because no named entity information was available, or because no pronouns with an unambiguous value for the property were present. For named entity information, OntoNotes only has annotations for 68% of gold entities, suggesting that named entity taggers are of limited usefulness, matching observations on the MUC and ACE corpora (Stoyanov et al., 2009).

The results in the 'Gold' column of Table 11 in-

Error type	Mentions			MUC			B^3		
	P	R	F	P	R	F	P	R	F
Span Error	2.8	2.8	2.7	2.8	2.8	2.8	1.0	2.0	1.6
Conflated Entities	1.7	0.0	0.8	9.9	0.0	4.5	**15.9**	0.0	6.2
Extra Mention	5.5	0.0	2.6	6.4	0.0	3.0	5.3	0.0	2.2
Extra Entity	**15.3**	0.0	7.0	**11.4**	0.0	5.2	6.1	0.0	2.4
Divided Entity	1.8	6.8	4.3	5.7	**16.8**	**10.9**	-10.0	**21.6**	4.5
Missing Mention	1.8	7.0	4.4	3.2	9.2	6.1	-1.3	7.3	3.4
Missing Entity	3.8	**16.2**	**9.8**	5.3	13.7	9.3	1.7	11.4	**7.0**

Table 10: Average accuracy improvement if all errors of a particular type are corrected. Each row in the lower section is calculated independently, relative to the change after the span errors have been corrected. Some values are negative because the merge operations involved in fixing the errors are applying to clusters that contain mentions from more than one gold entity.

Property	System	Gold
Named Entity	1.7%	0.7%
Gender	0.8%	0.1%
Number	2.1%	0.8%
Person	6.4%	5.1%

Table 11: Percentage of entities that contain mentions with properties that disagree.

dicate possible errors in the annotations, e.g. in the 0.7% of entities with a mixture of named entity types there may be mistakes in the coreference annotations, or mistakes in the named entity annotations.[3] However, even after taking into consideration cases where the mixture is valid and cases of annotation errors, current systems are placing mentions with different properties in the same clusters.

6.7 Impact of Errors on Metric Scores

Table 10 shows the performance impact of correcting errors of each type. The Span Error row gives improvements over the original scores, while all other rows are relative to the scores after Span Errors are corrected.[4] By fixing each of the other error types in isolation, we can get a sense of the gain if just that error type is addressed. However, it also means some mentions are incorrectly placed in the same cluster, causing some negative scores.

Interaction between the error types and the way the metrics are defined means that the deltas do not add up to the overall average gap in performance, but it is still clear that every error type has a noticeable impact. Missing Entity errors have the most substantial impact, which reflects the precision oriented nature of many coreference resolution systems.

7 Conclusion

While the improvement of metrics and the organization of shared tasks have been crucial for progress in coreference resolution, there is much insight to be gained by performing a close analysis of errors.

We have presented a new means of automatically classifying coreference errors that provides an exhaustive view of error types. Using our tool we have analyzed the output of a large set of coreference resolution systems and investigated the common challenges across state-of-the-art systems.

We find that there is considerable variability in the distribution of errors, and the best systems are not best across all error types. No single source of errors stands out as the most substantial challenge today. However, it is worth noting that while confidence measures can be used to reduce precision-related errors, no system has been able to effectively address the recall-related errors, such as Missed Entities. Our analysis tool is available at code.google.com/p/berkeley-coreference-analyser/.

Acknowledgments

We would like to thank the CoNLL task organizers for providing us with system outputs. This work was supported by a General Sir John Monash fellowship to the first author and by BBN under DARPA contract HR0011-12-C-0014.

[3] This kind of cross-annotation analysis may be a useful way of detecting annotation errors.

[4] This difference was necessary as the later errors make changes relative to the state of the entities after the Span Errors are corrected, e.g. in Figure 2 a blue and red entity is split that previously contained an X instead of one of the blue mentions.

References

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

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

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

Anders Björkelund and Pierre Nugues. 2011. Exploring lexicalized features for coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 45–50.

Jie Cai, Eva Mujdricza-Maydt, and Michael Strube. 2011. Unrestricted coreference resolution via global hypergraph partitioning. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 56–60.

Kai-Wei Chang, Rajhans Samdani, Alla Rozovskaya, Nick Rizzolo, Mark Sammons, and Dan Roth. 2011. Inference protocols for coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 40–44.

Eric Charton and Michel Gagnon. 2011. Poly-co: a multilayer perceptron approach for coreference detection. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 97–101.

Chen Chen and Vincent Ng. 2012. Combining the best of two worlds: A hybrid approach to multilingual coreference resolution. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 56–63.

Weipeng Chen, Muyu Zhang, and Bing Qin. 2011. Coreference resolution system using maximum entropy classifier. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 127–130.

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*.

Aria Haghighi and Dan Klein. 2009. 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.

Gordana Ilic Holen. 2013. Critical reflections on evaluation practices in coreference resolution. In *Proceedings of the 2013 NAACL HLT Student Research Workshop*, pages 1–7.

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.

Joseph Irwin, Mamoru Komachi, and Yuji Matsumoto. 2011. Narrative schema as world knowledge for coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 86–92.

Manfred Klenner and Don Tuggener. 2011. An incremental model for coreference resolution with restrictive antecedent accessibility. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 81–85.

Hamidreza Kobdani and Hinrich Schuetze. 2011. Supervised coreference resolution with sucre. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 71–75.

Jonathan K. Kummerfeld, Mohit Bansal, David Burkett, and Dan Klein. 2011. Mention detection: Heuristics for the ontonotes annotations. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 102–106.

Jonathan K. Kummerfeld, David Hall, James R. Curran, and Dan Klein. 2012. Parser showdown at the wall street corral: An empirical investigation of error types in parser output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1048–1059.

Sobha Lalitha Devi, Pattabhi Rao, Vijay Sundar Ram R, M. C S, and A. A. 2011. Hybrid approach for coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 93–96.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanfords multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34.

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

Xinxin Li, Xuan Wang, and Shuhan Qi. 2011. Coreference resolution with loose transitivity constraints.

In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 107–111.

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, Jie Cai, Samuel Broscheit, Éva Mújdricza-Maydt, and Michael Strube. 2012. A multigraph model for coreference resolution. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 100–106.

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

Cicero Nogueira dos Santos and Davi Lopes Carvalho. 2011. Rule and tree ensembles for unrestricted coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 51–55.

C. D. Paice and G. D. Husk. 1987. Towards the automatic recognition of anaphoric features in english text: the impersonal pronoun 'it'. *Computer Speech & Language*, 2(2):109–132.

Sameer Pradhan, Lance Ramshaw, Ralph Weischedel, Jessica MacBride, and Linnea Micciulla. 2007. Unrestricted coreference: Identifying entities and events in ontonotes. In *Proceedings of the International Conference on Semantic Computing*, pages 446–453.

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 15th Conference on Computational Natural Language Learning (CoNLL 2011)*, pages 1–27.

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.

Altaf Rahman and Vincent Ng. 2009. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 968–977.

M. Recasens and E. Hovy. 2011. BLANC: Implementing the rand index for coreference evaluation. *Natural Language Engineering*, 17:485–510, 9.

Emili Sapena, Lluís Padró, and Jordi Turmo. 2011. Relaxcor participation in conll shared task on coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 35–39.

Lalitha Devi. Sobha, RK. Rao. Pattabhi, R. Vijay Sundar Ram, CS. Malarkodi, and A. Akilandeswari. 2011. Hybrid approach for coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 93–96.

Yang Song, Houfeng Wang, and Jing Jiang. 2011. Link type based pre-cluster pair model for coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 131–135.

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.

Veselin Stoyanov, Claire Cardie, Nathan Gilbert, Ellen Riloff, David Buttler, and David Hysom. 2010. Coreference resolution with reconcile. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 156–161.

Veselin Stoyanov, Uday Babbar, Pracheer Gupta, and Claire Cardie. 2011. Reconciling ontonotes: Unrestricted coreference resolution in ontonotes with reconcile. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 122–126.

Olga Uryupina, Sriparna Saha, Asif Ekbal, and Massimo Poesio. 2011. Multi-metric optimization for coreference: The unitn / iitp / essex submission to the 2011 conll shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 61–65.

Yannick Versley, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti. 2008. Bart: a modular toolkit for coreference resolution. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Demo Session*, pages 9–12.

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

Hao Xiong, Linfeng Song, Fandong Meng, Yang Liu, Qun Liu, and Yajuan Lv. 2011. Ets: An error tolerable system for coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 76–80.

Yaqin Yang, Nianwen Xue, and Peter Anick. 2011. A machine learning-based coreference detection system

for ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 117–121.

Bo Yuan, Qingcai Chen, Yang Xiang, Xiaolong Wang, Liping Ge, Zengjian Liu, Meng Liao, and Xianbo Si. 2012. A mixed deterministic model for coreference resolution. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 76–82.

Desislava Zhekova and Sandra Kübler. 2011. Ubiu: A robust system for resolving unrestricted coreference. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 112–116.

Desislava Zhekova, Sandra Kübler, Joshua Bonner, Marwa Ragheb, and Yu-Yin Hsu. 2012. Ubiu for multilingual coreference resolution in ontonotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 88–94.

Huiwei Zhou, Yao Li, Degen Huang, Yan Zhang, Chunlong Wu, and Yuansheng Yang. 2011. Combining syntactic and semantic features by svm for unrestricted coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 66–70.