# Recall Error Analysis for Coreference Resolution

**Sebastian Martschat** and **Michael Strube**
Heidelberg Institute for Theoretical Studies gGmbH
Schloss-Wolfsbrunnenweg 35, 69118 Heidelberg, Germany
(sebastian.martschat|michael.strube)@h-its.org

## Abstract

We present a novel method for coreference resolution error analysis which we apply to perform a recall error analysis of four state-of-the-art English coreference resolution systems. Our analysis highlights differences between the systems and identifies that the majority of recall errors for nouns and names are shared by all systems. We characterize this set of common challenging errors in terms of a broad range of lexical and semantic properties.

## 1 Introduction

Coreference resolution is the task of determining which mentions in a text refer to the same entity. State-of-the-art approaches include both learning-based (Fernandes et al., 2012; Björkelund and Farkas, 2012; Durrett and Klein, 2013) and deterministic models (Lee et al., 2013; Martschat, 2013). These approaches achieve state-of-the-art performance mainly relying on morphosyntactic and lexical factors. However, consider the following example.

> In order to improving the added value of oil products, the second phase project of **the Qinghai Petroleum Bureau's Ge'ermu oil refinery** has been put into production. This will further improve **the factory's** oil products structure.

Due to the lack of any string overlap, most state-of-the-art systems will miss the link between *the factory* and *the Qinghai Petroleum Bureau's Ge'ermu oil refinery*. The information that *factory* is a hypernym of *refinery*, however, may be useful to resolve such links.

The aim of this paper is to quantify and characterize such recall errors made by state-of-the-art coreference resolution systems. By doing so, we provide a solid foundation for work on employing knowledge sources for improving recall for coreference resolution (Ponzetto and Strube, 2006; Rahman and Ng, 2011; Ratinov and Roth, 2012; Bansal and Klein, 2012, inter alia). In particular, we make the following contributions:

We present a novel framework for coreference resolution error analysis. This yields a formal foundation for previous work on link-based error analysis (Uryupina, 2008; Martschat, 2013) and complements work on transformation-based error analysis (Kummerfeld and Klein, 2013).

We apply the method proposed in this paper to perform a recall error analysis of four state-of-the-art systems, encompassing deterministic and learning-based approaches. In particular, we identify and characterize a set of challenging errors common to all systems, and discuss strengths and weaknesses of each system regarding specific error types. We also present a brief precision error analysis.

A toolkit which implements the framework proposed in this paper is available for download.[1]

## 2 A Link-Based Analysis Framework

In this section we discuss challenges in coreference resolution error analysis and devise an error analysis framework to overcome these challenges.

### 2.1 Motivation

Suppose a document contains the entity BARACK OBAMA, which is referenced by four mentions in the following order: *Obama*, *he*, *the president* and *his*. A typical output of a current system not equipped with world knowledge will consist of two entities: {*Obama*, *he*} and {*the president*, *his*}

Obviously, the system made a recall error. But, due to the complex nature of the coreference resolution task, it is not clear how to represent the re-

[1] http://smartschat.de/software



*Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2070–2081,
October 25-29, 2014, Doha, Qatar. ©2014 Association for Computational Linguistics

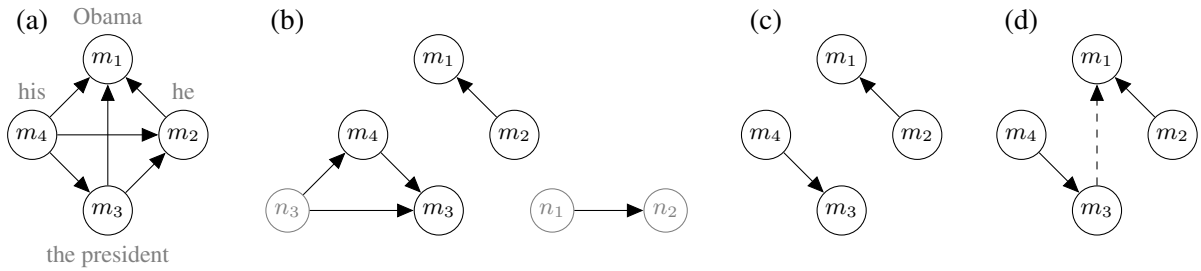


Figure 1: (a) a reference entity $r$, represented as a complete one-directional graph, (b) a set $S$ of three system entities, (c) the partition $r_S$, (d) a spanning tree for $r$.

call error: is it missing the link between *the president* and *Obama*? Can the error be attributed to deficiencies in pronoun resolution?

Linguistically motivated error representations would facilitate both understanding of current challenges and make system development faster and easier. The aim of this section is to devise such representations.

### 2.2 Formalizing Coreference Resolution

To start with, we give a formal description of the coreference resolution task following the terminology used for the ACE (Mitchell et al., 2004) and OntoNotes (Weischedel et al., 2013) projects. A *mention* is a linguistic realization of a reference to an entity. Two mentions *corefer* if they refer to the same entity. Hence, coreference is reflexive, symmetric and transitive, and therefore an *equivalence relation*. The task of coreference resolution is to predict equivalence classes of mentions in a document according to the coreference relation.

In order to extract errors, we need to compare the *reference equivalence classes*, given by the annotation, with the *system equivalence classes* obtained from system output. The key question now is how we represent these equivalence classes of mentions. Adapting common terminology, we also refer to the equivalence classes as entities.

### 2.3 Representing Entities

The most straightforward entity representation ignores any structure and models an entity as a set of mentions. This representation was utilized for error analysis by Kummerfeld and Klein (2013), who extract errors by transforming reference into system entities. In this set-based representation, we can only extract whether two mentions corefer at all. More fine-grained information, for example about antecedent information, is not accessible.

We therefore propose to employ a *structured* entity representation, which explicitly models links established by the coreference relation between mentions. This leads to a link-based error representation which formalizes the methods presented in Uryupina (2008) and Martschat (2013).

We employ for representation a *complete one-directional graph*. That is, we represent an entity $e$ over mentions $\{m_1, \ldots, m_n\}$ as a graph $e = (N, A)$, where $N = \{m_1, \ldots, m_n\}$ and $A = \{(m_k, m_j) \mid k > j\}$. The indices respect the mention ordering. Mentions earlier in the text have a lower index. An example graph for an entity over four mentions $m_1, \ldots, m_4$ (such as the BARACK OBAMA entity) is depicted in Figure 1a. In this graph, we express all coreference relations between all pairs of mentions.[2]

Using this representation, we can represent a set of entities as a set of graphs. In particular, given a document we consider the set of *reference entities* $R$ given by the annotation, and the set of *system entities* $S$, given by the system output. In order to extract errors, we compare the graphs in $R$ with the graphs in $S$.

In the following, we discuss how to compute recall errors for a reference entity $r \in R$ with respect to the system entities $S$. For computing precision errors, we just switch the roles of $R$ and $S$.

### 2.4 Comparing Reference and System Entities

As we represent entities as sets of links between mentions, errors can be quantified as differences in the links. For example, if an edge (representing a link) from some reference entity $r \in R$ is missing

[2]We could also use an undirected instead of a one-directional graph, but using a one-directional graph conveniently models sequential information, which simplifies notation and the algorithms we will present.

in all system entities in $S$, this is a recall error.

In order to formalize this, we employ the notion of a *partition* of an entity. Let $r \in R$ be some reference entity, and let $S$ be a set of system entities. The partition of $r$ by $S$, written $r_S$, is obtained by taking all edges in $r$ that also appear in $S$. $r_S$ consists of all connected components of $r$ (we will refer to these as *subentities*) that are also in $S$. All edges in $r$ that are not in $r_S$ are candidates for recall errors, as these were not in any entity in $S$.

Figure 1b shows a set $S$ of three system entities: two consist of two mentions, one of three mentions. In our running example, this corresponds to the system output {*Obama*, *he*} and {*the president*, *his*} plus some spurious mentions, which are colored gray. The graph $r_S$ for our example is shown in Figure 1c. The two edges correspond to the correctly recognized links (*he*, *Obama*) and (*his*, *the president*). All edges in $r$ (Figure 1a) missing from this graph are candidates for errors.

### 2.5 Spanning Trees

However, taking all edges in $r$ missing in $r_S$ as errors leads to unintuitive results. In the BARACK OBAMA example, this would lead to four errors being extracted: (*the president*, *Obama*), (*his*, *Obama*), (*the president*, *he*) and (*his*, *he*). But, in order to correctly predict the BARACK OBAMA entity, a coreference resolution system only needs to predict three correct links, i.e. it has to provide a *spanning tree* of the entity's graph representation.

Therefore, to extract errors, we compute a spanning tree $T_r$ of $r$, and take all edges in $T_r$ that do not appear in $r_S$ as errors. Figure 1d shows an example spanning tree for the running example entity $r$. The dashed edge, which corresponds to the link (*the president*, *Obama*), does not appear in $r_S$ and is therefore extracted as an error.

The strategies for computing a spanning tree may differ for recall and precision errors. Hence, our extraction algorithm is parametrized by two procedures $\texttt{ST}_{\text{rec}}(e, P)$ and $\texttt{ST}_{\text{prec}}(e, P)$ which, given an entity $e$ and a set of entities $P$, output a spanning tree $T_e$ of $e$. The whole algorithm for error extraction is summarized in Algorithm 1.

**Algorithm 1** Error Extraction from a Corpus

**Input:** A corpus $\mathcal{C}$, algorithms $\texttt{ST}_{\text{rec}}$, $\texttt{ST}_{\text{prec}}$ for computing spanning trees.

**function** ERRORS($\mathcal{C}$, $\texttt{ST}_{\text{rec}}$, $\texttt{ST}_{\text{prec}}$)

`recall_errors` $= [\,]$

`precision_errors` $= [\,]$

**for** $d \in \mathcal{C}$ **do**

Let $R_d$ be the reference entities and $S_d$ be the system entities of document $d$.

**for** $r \in R_d$ **do**

Add all edges in $\texttt{ST}_{\text{rec}}(r, S_d)$ not in $r_{S_d}$ to `recall_errors`.

**for** $s \in S_d$ **do**

Add all edges in $\texttt{ST}_{\text{prec}}(s, R_d)$ not in $s_{R_d}$ to `precision_errors`.

**Output:** `recall_errors`, `precision_errors`

## 3 Spanning Tree Algorithms

In the last section we presented a framework for link-based error analysis, which extracts errors by comparing entity spanning trees to entity partitions. Therefore we can accommodate different notions of errors by varying the algorithm for computing spanning trees. We now present some spanning tree algorithms for extracting recall and precision errors.

### 3.1 Recall Errors

We first observe that for error extraction, the structure of the spanning trees of the subentities appearing in $r_S$ does not play a role. Edges present in $r_S$ are not candidates for errors, since they appear in both the reference entity $r$ and the system output $S$. Therefore, it does not matter which edges from the subentities are in the spanning tree.

Hence, to build the spanning tree, we first choose arbitrary spanning trees for the subentities in the partition. We choose the remaining edges according to the spanning tree algorithm.

Having settled on this, we only have to decide which edges to choose that connect the trees representing the subentities. There are many possible choices for this. For example, the graph in Figure 1c has four candidate edges which connect the trees for the subentities.

We can reduce the number of candidate edges by only considering the first mention (with respect to textual order) in a subentity as the source of an edge to be added. This makes sense since all other mentions in that subentity were correctly resolved to be coreferent with some preceding mention. We still have to decide on the target of the edge. In Figure 1c, we have two choices for edges: $(m_3, m_1)$ and $(m_3, m_2)$. We now present two methods for choosing edges.

**Choosing Edges by Distance.** The most straight-forward way to decide on an edge is to take the edge with smallest mention distance between source and target. This is the approach taken by Martschat (2013).

**Choosing Edges by Accessibility.** However, the distance-based approach may lead to unintuitive results. Let us consider again the BARACK OBAMA example from Figure 1. When choosing edges by distance, we would extract the error (*the president*, *he*). However, such links with a non-pronominal anaphor and a pronominal antecedent are difficult to process and considered unreliable (Ng and Cardie, 2002; Bengtson and Roth, 2008). On the other hand, the missed link (*the president*, *Obama*) constitutes a well-defined hyponymy relation which can be found in knowledge bases and is easily interpretable by humans.

Uryupina (Uryupina, 2007; Uryupina, 2008) presents a recall error analysis where she takes the "intuitively easiest" missing link to analyze (Uryupina, 2007, p. 196). How can we formalize such an intuition? We will employ a notion grounded in accessibility theory (Ariel, 1988). Names and nouns refer to less accessible entities than pronouns do. For such anaphors, we prefer descriptive (name/nominal) antecedents. Inspired by Ariel's degrees of accessibility, we choose a target for a given anaphor $m_i$ as follows:

- If $m_i$ is a pronoun, choose the closest preceding mention.
- If $m_i$ is not a pronoun, choose the closest preceding proper name. If no such mention exists, choose the closest preceding common noun. If no such mention exists, choose the closest preceding mention.

Applied to the example from Figure 1, this algorithm extracts the error (*the president*, *Obama*).[3]

### 3.2 Precision Errors

Virtually all approaches to coreference resolution obtain entities by outputting pairs of anaphor and antecedent, subject to the constraint that one anaphor has at most one antecedent.

We use this information to build spanning trees for system entities: these spanning trees consist of exactly the edges which correspond to anaphor/antecedent pairs in the system output.

[3] A similar procedure was used by Ng and Cardie (2002) to extract meaningful antecedents when training a coreference resolution system.

## 4 Data and Systems

We now discuss data and coreference resolution systems which we will employ for our analysis.

### 4.1 Data

We analyze the errors of the systems on the English development data of the CoNLL'12 shared task on multilingual coreference resolution (Pradhan et al., 2012). This corpus contains 343 documents, spanning seven genres: bible texts, broadcast conversation, broadcast news, magazine texts, news wire, telephone conversations and web logs.

### 4.2 Systems

State-of-the-art approaches to coreference resolution encompass various paradigms, ranging from deterministic pairwise systems to learning-based structured prediction models. Hence, we want to conduct our analysis on a representative sample of the state of the art, which should be publicly available. Therefore, we decided on two deterministic and two learning-based systems:

- **StanfordSieve**[4] (Lee et al., 2013) was the winning system of the CoNLL'11 shared task. It employs a multi-sieve approach by making more confident decisions first.
- **Multigraph**[5] (Martschat, 2013) is a deterministic pairwise system which is based on Martschat et al. (2012), the second-ranking system in the English track of the CoNLL'12 shared task. It uses a subset of features as hard constraints and chooses an antecedent for a mention by summing up the remaining boolean features.
- **IMSCoref**[6] (Björkelund and Farkas, 2012) ranked second overall in the CoNLL'12 shared task (third for English). It stacks multiple decoders and relies on a combination of standard pairwise and lexicalized features.
- **BerkeleyCoref**[7] (Durrett and Klein, 2013) is a state-of-the-art system that uses mainly lexicalized features and a latent antecedent ranking architecture. It outperforms StanfordSieve and IMSCoref on the CoNLL'11 data.

[4] Part of Stanford CoreNLP, available at `http://nlp.stanford.edu/software/corenlp.shtml`. We use version 3.4.

[5] `http://smartschat.de/software`

[6] `http://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/IMSCoref.en.html` . We use the *CoNLL 2012 system*.

[7] `http://nlp.cs.berkeley.edu/berkeleycoref.shtml`

| System | MUC | $B^3$ | $CEAF_e$ | Average |
|---|---|---|---|---|
| Fernandes et al. | 69.46 | 57.83 | 54.43 | 60.57 |
| Martschat | 66.22 | 55.47 | 51.90 | 57.86 |
| StanfordSieve | 64.96 | 54.49 | 51.24 | 56.90 |
| Multigraph | 69.13 | 58.61 | 56.06 | 61.28 |
| IMSCoref | 67.15 | 55.19 | 50.94 | 57.76 |
| BerkeleyCoref | 70.27 | 59.29 | 56.11 | 61.89 |

Table 1: Comparison of the systems with Fernandes et al. (2012) and with Martschat (2013) on CoNLL'12 English development data.

For Multigraph, we modified the system described in Martschat (2013) slightly to allow for the incorporation of distance (similar to Cai and Strube (2010)). Inspired by Lappin and Leass (1994), we add salience weights for subjects and objects to the model to improve third-person pronoun resolution. We also extended the feature set by a substring feature. Furthermore, motivated by Chen and Ng (2012), we added a lexicalized feature for non-pronominal mentions that were coreferent in at least 50% of the cases in the training data.

StanfordSieve was run with its standard CoNLL shared task settings. The learning-based systems were trained on the CoNLL'12 training data. We trained IMSCoref with its standard settings, and trained BerkeleyCoref with the *final* feature set from Durrett and Klein (2013) for twenty iterations. We evaluate the systems on English CoNLL'12 development data and compare it with the winning system of the CoNLL'12 shared task (Fernandes et al., 2012) and with Martschat (2013) in Table 1, using the reference implementation v7 of the CoNLL scorer (Pradhan et al., 2014).

BerkeleyCoref performs best according to all metrics, followed by Multigraph. StanfordSieve is the worst performing system: the gap to BerkeleyCoref is five points in average score.

### 4.3 Discussion

Although we analyze recent systems on a recently published coreference data set, we believe that the results of our analysis will have implications for coreference in general. The data set is the largest and most genre-diverse coreference corpus so far. The systems we investigate represent major directions in coreference resolution model research, and make use of large and diverse feature sets proposed in the literature (Ng, 2010).

## 5 A Comparative Analysis

The coreference resolution systems presented in the previous section are a representative sample of the state of the art. Therefore, by analyzing the errors they make, we can learn about remaining challenges in coreference resolution and analyze the qualitative differences between the systems. The results of such an analysis will deepen our understanding of coreference resolution and will suggest promising directions for further research.

### 5.1 Experimental Settings

Previous studies identified the presence of recall errors as a main bottleneck for improving performance (Raghunathan et al., 2010; Durrett and Klein, 2013; Kummerfeld and Klein, 2013). This is also evidenced by the CoNLL shared tasks on coreference resolution (Pradhan et al., 2011; Pradhan et al., 2012), where most competitive systems had higher precision than recall. This indicates that an analysis of recall errors helps to understand and improve the state of the art. Hence, we focus on analyzing recall errors, and complement this by a brief analysis of precision errors.

We analyze errors of the four systems presented in the previous section on the CoNLL'12 English development data. To extract recall errors we employ the spanning tree algorithm which **chooses edges by accessibility**. We obtain precision errors from the pairwise output of the systems.

### 5.2 A Recall Error Analysis of StanfordSieve

Since StanfordSieve is currently the most-widely used coreference resolution system, it serves as a good starting point for our analysis. Remember that we represent each error as a pair of anaphor and antecedent. For an initial analysis, we categorize each error by mention type, distinguishing between proper name, common noun, pronoun, demonstrative pronoun and verb.[8]

StanfordSieve makes 5245 recall errors. To put this number into context, we compare it with the maximum number of recall errors a system can make. This count is obtained by extracting recall errors from the output of a system that puts each mention in its own entity, which yields 14609 errors. In Table 2 we present a detailed analysis. For each pair of mention type of anaphor and an-

[8]We obtain the type from the part-of-speech tag of the mention's head. Furthermore, we treat every mention whose head has a NER label in the data as a proper name.

| | Name | Noun | Pron. | Dem. | Verb |
|---|---|---|---|---|---|
| **Name** | | | | | |
| Errors | 1006 | 181 | 43 | 0 | 0 |
| Maximum | 3578 | 206 | 56 | 2 | 0 |
| **Noun** | | | | | |
| Errors | 517 | 1127 | 46 | 14 | 91 |
| Maximum | 742 | 2063 | 51 | 14 | 91 |
| **Pron.** | | | | | |
| Errors | 483 | 761 | 543 | 45 | 53 |
| Maximum | 1166 | 1535 | 4596 | 92 | 53 |
| **Dem.** | | | | | |
| Errors | 23 | 86 | 41 | 31 | 117 |
| Maximum | 27 | 93 | 43 | 46 | 117 |
| **Verb** | | | | | |
| Errors | 1 | 20 | 2 | 4 | 10 |
| Maximum | 1 | 20 | 2 | 4 | 11 |

Table 2: Number of StanfordSieve's recall errors according to mention type, compared to the maximum possible number of errors. Rows are anaphors, columns antecedents.

tecedent, the table displays the number of recall errors and of maximum errors possible.

StanfordSieve gets almost none of the links involving verbal or demonstrative mentions correct. This is due to the system not attempting to handle event coreference, and performing very poorly for demonstratives. On the other hand, recall for pronoun resolution is quite good, at least when considering non-verbal antecedents. While StanfordSieve makes 1885 recall errors when the anaphor is a pronoun, it successfully resolves most of such links present in the corpus. Finally, let us consider the links involving only proper names and common nouns. In total, these amount to 6589 links in the corpus (around 45% of all links). StanfordSieve misses 2831 of these links. Pairs of proper names seem to be easier to resolve than pairs of common nouns. Links between a common noun and a proper name are less frequent, but much more difficult: most of the links are missing.

### 5.3 Analysis of the Other Systems

In the previous section we identified various characteristics of the errors made by StanfordSieve: only (comparatively) few errors are made for pronoun resolution and name coreference, while other types of nominal anaphora and coreference of demonstrative/verbal mentions pose a challenge for the system. Do the other systems in our study also have these characteristics? In order to answer this question, we repeated the analysis for the three other systems described in Section 4. We summarize the results in Table 3. We only report numbers for pronoun resolution and name/noun coreference, as all systems do not resolve verbal mentions and perform poorly for demonstratives.

| | | Proportion | |
|---|---|---|---|
| **System** | **Total** | **Anaphor Pron.** | **Name/Noun** |
| **StanfordSieve** | 5245 | 36% | 54% |
| **Multigraph** | 4630 | 32% | 56% |
| **IMSCoref** | 5220 | 32% | 58% |
| **BerkeleyCoref** | 4635 | 32% | 56% |

Table 3: Recall error numbers for all systems.

StanfordSieve makes the most recall errors, closely followed by IMSCoref. Multigraph and BerkeleyCoref make around 600 errors less. While the total number of errors differs between the systems, the distributions are similar. In particular, around 55% of recall errors made involve only proper names and common nouns. The number is a bit higher for IMSCoref. We conclude that, despite variations in performance, both deterministic and learning-based state-of-the-art systems have similar weaknesses regarding recall.

The results displayed in Table 3 suggest various opportunities for future research. In this paper, we will focus on analyzing name/noun recall errors, as these constitute a large fraction of all recall errors. Future work should address the pronoun resolution errors and a characterization of the verbal/demonstrative errors.

### 5.4 Analysis of the Name/Noun Recall Errors

We now turn towards a fine-grained analysis of the name/noun recall errors.

Table 4 displays the number of such recall errors made by each system, according to the mention types of anaphor and antecedent. We are interested in errors common to all systems, and in qualitative differences of errors between the systems.

#### 5.4.1 Common Errors

Let us first analyze the errors common to all systems. Our analysis is driven by the question how these can be characterized, and which knowledge is missing to resolve such links. We discuss the errors depending on the mention types of anaphor and antecedent. The lower part of Table 4 displays the number of common errors for each category.

| Description | Number of Recall Errors (Anaphor-Antecedent) | | | |
|---|---|---|---|---|
| | Name-Name | Noun-Name | Name-Noun | Noun-Noun |
| StanfordSieve | 1006 | 517 | 181 | 1127 |
| Multigraph | 753 | 501 | 189 | 1152 |
| IMSCoref | 1082 | 500 | 188 | 1264 |
| BerkeleyCoref | 910 | 456 | 171 | 1072 |
| Common errors | 475 | 371 | 147 | 835 |
| Correct boundaries idenfified | 257 | 273 | 108 | 563 |
| excl. IMSCoref | 156 | 222 | 97 | 475 |

Table 4: Name/Noun recall errors for all systems.

| Common | | All | |
|---|---|---|---|
| Type | % | Type | % |
| ORG | 25% | PERSON | 26% |
| PERSON | 19% | GPE | 26% |
| GPE | 16% | ORG | 20% |
| DATE | 14% | NONE | 14% |
| NONE | 9% | DATE | 6% |

Table 5: Distribution of top five named entity types of common name-name recall errors and all possible name-name recall errors.

Furthermore, in order to assess the impact of mention detection, the table shows the number of common errors where boundaries for both mentions were identified correctly by some system. We can see that boundary identification is a difficult problem, especially for proper name pairs: for 48% of such errors, no system found the correct boundaries of both mentions participating in the error. The number of errors where correct boundaries could be found drops significantly after excluding IMSCoref. This is due to the mention extraction strategy of IMSCoref: the other systems in our study discard the shorter mention when two mentions have the same head, IMSCoref keeps both mentions. Hence, the system is able to correctly identify some mentions even in the presence of parsing or preprocessing errors. However, as a result, IMSCoref has to process many spurious mentions, which makes learning more difficult.

We conclude that mention detection still constitutes a challenge. We now proceed to a detailed analysis of errors common to all systems. In passing we will discuss difficulties in mention detection with regard to specific error types.

**Errors between Pairs of Proper Names.** The systems share 475 recall errors between pairs of proper names. In Table 5, we compare the distribution of gold named entity types of these errors with the distribution of gold named entity types of all possible errors (obtained via a singleton system). We see that especially difficult classes of links are pairs with type ORG or DATE.

Let us now consider lexical features of the errors.[9] In 154 errors, the strings match completely, but the correct resolution was mostly prevented by annotation inconsistencies (e.g. *China* instead of *China's*) or propagated parsing and NER errors, which lead to deficiencies in mention extraction.

For 217 errors, at least one token appears in both mention strings, as in *the "Cole"* and *the "USS Cole"*. This shows the insufficiency of the features which hint to alias relations, may it be heuristics or learned lexical similarities (for 109 of the 217 errors, both mention boundaries were identified correctly by at least one system). Disambiguation with respect to knowledge bases could provide a principled way to identify name variations.

We classified the remaining 104 errors manually, see Table 6. For a couple of categories such as identifying acronyms, spelling variations and aliases, disambiguation could also help. Many errors happen for date mentions, which suggests the use of temporal tagging features.

**Errors for Noun-Name Pairs.** We now investigate the errors where the anaphor is a common noun and the antecedent is a proper name. 371 errors are common to all systems. The high fraction of common errors shows that this is an especially challenging category. We again start by investigating how the distribution of the named entity type

[9] When computing these, we ignored case and ignored all tokens with part-of-speech tag DT or POS.

| Description | Occ. | Example |
|---|---|---|
| Acronyms | 20 | *National Ice Hockey League* and *NHL* |
| Alias | 24 | *Florida* and *the Sunshine State* |
| Annotation | 2 | Annotation errors (pronoun as name) |
| Context | 2 | *Paula Coccoz* and *juror number ten* |
| Date | 29 | *1989* and *last year's* |
| Metonymy | 12 | *South Afria* and *Pretoria* |
| Roles | 8 | *Al Gore* and *the Vice President* |
| Spelling | 7 | *Hsiachuotzu* and *Hsiachuotsu* |

Table 6: Classification of common name-name recall errors without common tokens.

| Common | | All | |
|---|---|---|---|
| Type | % | Type | % |
| ORG | 28% | ORG | 27% |
| PERSON | 22% | GPE | 22% |
| GPE | 19% | PERSON | 18% |
| NONE | 7% | NONE | 11% |
| DATE | 5% | DATE | 5% |

Table 7: Distribution of top five named entity types of common noun-name recall errors and all possible noun-name recall errors.

of the antecedent differs when we compare common errors to all possible errors. The results are shown in Table 7. Links with a proper name antecedent of type PERSON are especially difficult. They constitute 22% of the common errors, but only 18% of all possible errors.

Most mentions are in a hyponymy relation, like *the prime minister* and *Mr. Papandreou*. This confirms that harnessing such relations could improve coreference resolution (Rahman and Ng, 2011; Uryupina et al., 2011). For 65 of the errors (18%) there is lexical overlap: the head of the anaphor is contained in the proper name antecedent, as in *the entire park* and *the Ocean Park*.

When categorizing all common errors according to the head of the anaphor, we observe 204 different heads. 142 heads appear only once, but the top ten heads make up 88 of the 371 errors. The most frequent heads are *company* (15), *group* (12), *government*, *country* and *nation* (each 9). This suggests that even with few reliable hyponymy relations recall could be significantly improved.

We observe similar trends when the anaphor is a proper name and the antecedent is a noun.

| | Reference System | | | |
|---|---|---|---|---|
| System | Stanford | MG | IMS | Berkeley |
| **Stanford** | - | 51 | 47 | 61 |
| **MG** | 17 | - | 42 | 60 |
| **IMS** | 26 | 54 | - | 54 |
| **Berkeley** | 12 | 42 | 25 | - |

Table 8: Comparison of noun-name recall errors. Entries are errors made by the system in the row, while the participating mentions are coreferent according to the the system in the column.

**Errors between Pairs of Common Nouns.** 835 errors between pairs of common nouns are shared by all systems. For 174 of these, the anaphor is an indefinite noun phase, which makes resolution a lot harder, since most coreference resolution systems classify these as non-anaphoric and therefore do not attempt resolution.

For further analysis, we split all 835 errors in two categories, distinguishing whether the head matches between the mentions or not. In 341 cases the heads match. For many of these cases, parsing errors propagate and prevent the systems from recognizing the correct mention boundaries.

In order to get a better understanding of the errors for nouns with different heads, we randomly extracted 50 of the 494 pairs and investigated the relation that holds between the heads. In 23 cases, the heads were related via hyponymy. In 10 cases they were synonyms. The remaining 17 cases involve many different phenomena, for example meronymy. This confirms findings from previous research (Vieira and Poesio, 2000).

Hence, looking up lexical relations, especially hyponymy, might be helpful to solve these cases.

### 5.4.2 Differences between the Systems

In order to analyze differences between the systems, we compare the recall errors they make. The information how recall errors differ between systems will enable us to understand individual strengths and weaknesses.

Exemplarily, we will have a look at the differences in the errors when the anaphor is a common noun and the antecedent is a proper name. By system design and by the total error numbers (Table 4) we expect the learning-based systems to have a slight advantage over the deterministic systems.

In Table 8 we compare noun-name recall errors made by each system. Entries are errors made by

| | Number and Proportion of Precision Errors (Anaphor-Antecedent) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Description** | **Name-Name** | | **Noun-Name** | | **Name-Noun** | | **Noun-Noun** | |
| StanfordSieve | 1038 | 31% | 64 | 59% | 65 | 72% | 944 | 48% |
| Multigraph | 1131 | 30% | 76 | 51% | 24 | 56% | 743 | 42% |
| IMSCoref | 834 | 26% | 74 | 59% | 46 | 64% | 1050 | 54% |
| BerkeleyCoref | 810 | 24% | 191 | 67% | 60 | 62% | 1015 | 48% |
| Common errors | 158 | | 1 | | 2 | | 167 | |

Table 9: Name/Noun precision errors for all systems. The percentages are the proportion of precision errors with respect to all decision of the system in that category.

the system in the row, while the participating mentions are coreferent according to the the system in the column. The numbers confirm our hypothesis, but also show that the deterministic systems are able to recover a few links missed by the learning-based systems.

For example, BerkeleyCoref recovers 60 links that could not be found by Multigraph, including 34 links without any common token, such as *the airline* and *Pan Am*. Multigraph recovers only 42 links not found by BerkeleyCoref, 21 without any common token. Qualitatively, StanfordSieve and Multigraph are able to resolve a few links thanks to their engineered substring match, such as *the judge* and *Dallas District Judge Jack Hampton*.

We also conducted similar investigations for common noun and proper name pairs. For common nouns, the trends are similar: the learning-based systems have an advantage over the deterministic systems. However, only few relations between nouns with different heads are learned – compared to StanfordSieve, BerkeleyCoref recovers only 11 such pairs, such as *the man* and *an expert in the law*. Recall of the deterministic systems is further hampered by their strict checks for modifier agreement, which they employ to keep precision high. Both systems miss for example the link from the anaphor *the Milosevic regime* to *the regime*, since the nominal modifier of the anaphor does not appear in the antecedent.

For proper names, Multigraph employs sophisticated alias heuristics which help to resolve matches such as *Marshall Ye Ting's* and *his grandfather Ye Ting*. This explains the corresponding low number in Table 4. The lexicalized features of Multigraph, IMSCoref and BerkeleyCoref help to learn aliases when there is no string match, especially for the bible part of the corpus (resolving links such as *Jesus* and *the Son of Man*).

### 5.5 Precision Errors

In the above analysis we identified common name/noun recall errors and discussed strengths and weaknesses of each system. Let us complement this analysis by a brief discussion of corresponding precision errors.

Table 9 gives an overview. It displays the number of precision errors for each category, and the proportion of these errors compared to all decisions in that category. We can see some general trends from this table: first, more decisions lead to a higher proportion of errors. This shows the difficulty of balancing recall and precision. Second, proper name coreference seems much easier than common noun coreference. Coreference involving different mention types is a lot harder – the systems only attempt few decisions, most of them are wrong. This confirms findings from our recall error analysis. Third, the fraction of common errors is very low, which indicates that precisions errors stem from various sources, which are handled differently by each system.

## 6 Related Work

We now discuss related work in coreference resolution error analysis and in the related field of coreference resolution evaluation metrics.

**Error Analysis.** While many papers on coreference resolution briefly discuss errors made and resolved by the system under consideration, only few concentrate on error analysis. Uryupina (2008) presents a manual error analysis on the small MUC-7 test set; Martschat (2013) performs an automatic coarse-grained error classification on CoNLL data. By extending and formalizing the approach of Martschat (2013), we are able to perform a large-scale investigation of recall errors made by state-of-the-art systems.

Kummerfeld and Klein (2013) devise a method to extract errors from transformations of reference to system entities. They apply this method to a variety of systems and aggregate errors over these systems. By aggregating, they are not able to analyze differences. They furthermore focus on describing many different error classes, instead of closely investigating particular phenomena.

**Evaluation Metrics.** We extract recall and precision errors. How does our error analysis framework relate to coreference resolution evaluation metrics, which *quantify* recall and precision errors? We first observe a fundamental difference: evaluation metrics deal with *scoring* coreference chains, they provide no means of extracting recall or precision errors. Therefore our analysis complements insights obtained via evaluation metrics.

We follow Chen and Ng (2013) and distinguish between *linguistically agnostic metrics*, which do not employ linguistic information during scoring, and *linguistically informed metrics*, which employ linguistic information similar as we do when computing spanning trees.

We limit the discussion of linguistically agnostic metrics to the three most popular evaluation metrics whose average constitutes the official score in the CoNLL shared tasks on coreference resolution: MUC (Vilain et al., 1995), $B^3$ (Bagga and Baldwin, 1998) and $CEAF_e$ (Luo, 2005).[10]

Our framework bears most similarities to the MUC metric, as both are based on the same link-based entity representation. In particular, when we divide the number of errors extracted from an entity by the size of a spanning tree for that entity, we obtain a score linearly related[11] to the MUC score for that entity (recall for reference entities, precision for system entities). $B^3$ and $CEAF_e$ are not founded on a link-based structure. $B^3$ computes recall by computing the relative overlap of reference and system entity for each reference mention, and then normalizes by the number of mentions. $CEAF_e$ computes an optimal entity alignment with respect to the relative overlap, and then normalizes by the number of entities. As the metrics are not link-based, they do not provide means to extract link-based errors. We leave determining whether the framework of these metrics exhibits a useful notion of errors to future work.

[10]These are linguistically agnostic since they do not differ between different mention or entity types when evaluating.

[11]via the transformation $x \mapsto 1 - x$

Recent work considered devising evaluation metrics which take linguistic information into account. Chen and Ng (2013) inject linguistic knowledge into existing evaluation metrics by weighting links in an entity representation graph. Tuggener (2014) devises scoring algorithms tailored for particular applications by redefining the notion of a correct link. While both of these works focus on scoring, they weight or explicitly define links in the reference and system entities, thereby they in principle allow error extraction. However, the authors do not attempt this and it is not clear whether the errors extracted that way are useful for analysis and system development.

## 7 Conclusions

We presented a novel link-based framework for coreference resolution error analysis, which extends and complements previous work. We applied the framework to analyze recall errors of four state-of-the-art systems on a large English benchmark dataset. Concentrating on errors involving only proper names and common nouns, we identified a core set of challenging errors common to all systems in our study.

We characterized the common errors among a broad range of properties. In particular, our analysis highlights and quantifies the usefulness of world knowledge. Furthermore, by comparing the recall errors made by each system, we identified individual strengths and weaknesses. A brief precision error analysis highlighted the hardness of resolving noun-name and noun-noun links.

The presented method and findings help to identify challenges in coreference resolution and to investigate ways to overcome these challenges.

## Acknowledgments

This work has been funded by the Klaus Tschira Foundation, Heidelberg, Germany. The first author has been supported by a HITS Ph.D. scholarship.

## References

Mira Ariel. 1988. Referring and accessibility. *Journal of Linguistics*, 24(1):65–87.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the 1st International Conference on Language Resources and Evaluation,* Granada, Spain, 28–30 May 1998, 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),* Jeju Island, Korea, 8–14 July 2012, pages 389–398.

Eric Bengtson and Dan Roth. 2008. Understanding the value of features for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing,* Waikiki, Honolulu, Hawaii, 25–27 October 2008, pages 294–303.

Anders Björkelund and Richárd Farkas. 2012. Data-driven multilingual coreference resolution using resolver stacking. In *Proceedings of the Shared Task of the 16th Conference on Computational Natural Language Learning,* Jeju Island, Korea, 12–14 July 2012, pages 49–55.

Jie Cai and Michael Strube. 2010. End-to-end coreference resolution via hypergraph partitioning. In *Proceedings of the 23rd International Conference on Computational Linguistics,* Beijing, China, 23–27 August 2010, pages 143–151.

Chen Chen and Vincent Ng. 2012. Combining the best of two worlds: A hybrid approach to multilingual coreference resolution. In *Proceedings of the Shared Task of the 16th Conference on Computational Natural Language Learning,* Jeju Island, Korea, 12–14 July 2012, pages 56–63.

Chen Chen and Vincent Ng. 2013. Linguistically aware coreference evaluation metrics. In *Proceedings of the 6th International Joint Conference on Natural Language Processing,* Nagoya, Japan, 14–18 October 2013, pages 1366–1374.

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,* Seattle, Wash., 18–21 October 2013, pages 1971–1982.

Eraldo Fernandes, Cícero dos Santos, and Ruy Milidiú. 2012. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *Proceedings of the Shared Task of the 16th Conference on Computational Natural Language Learning,* Jeju Island, Korea, 12–14 July 2012, pages 41–48.

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,* Seattle, Wash., 18–21 October 2013, pages 265–277.

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

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

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of the Human Language Technology Conference and the 2005 Conference on Empirical Methods in Natural Language Processing,* Vancouver, B.C., Canada, 6–8 October 2005, pages 25–32.

Sebastian Martschat, Jie Cai, Samuel Broscheit, Éva Mújdricza-Maydt, and Michael Strube. 2012. A multigraph model for coreference resolution. In *Proceedings of the Shared Task of the 16th Conference on Computational Natural Language Learning,* Jeju Island, Korea, 12–14 July 2012, pages 100–106.

Sebastian Martschat. 2013. Multigraph clustering for unsupervised coreference resolution. In *51st Annual Meeting of the Association for Computational Linguistics: Proceedings of the Student Research Workshop,* Sofia, Bulgaria, 5–7 August 2013, pages 81–88.

Alexis Mitchell, Stephanie Strassel, Shudong Huang, and Ramez Zakhary. 2004. ACE 2004 multilingual training corpus. LDC2005T09, Philadelphia, Penn.: Linguistic Data Consortium.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics,* Philadelphia, Penn., 7–12 July 2002, pages 104–111.

Vincent Ng. 2010. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,* Uppsala, Sweden, 11–16 July 2010, pages 1396–1411.

Simone Paolo Ponzetto and Michael Strube. 2006. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics,* New York, N.Y., 4–9 June 2006, pages 192–199.

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 Shared Task of the 15th Conference on Computational Natural Language Learning,* Portland, Oreg., 23–24 June 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 *Proceedings*

*of the Shared Task of the 16th Conference on Computational Natural Language Learning,* Jeju Island, Korea, 12–14 July 2012, pages 1–40.

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),* Baltimore, Md., 22–27 June 2014, pages 30–35.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing,* Cambridge, Mass., 9–11 October 2010, pages 492–501.

Altaf Rahman and Vincent Ng. 2011. Coreference resolution with world knowledge. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Portland, Oreg., 19–24 June 2011, pages 814–824.

Lev Ratinov and Dan Roth. 2012. Learning-based multi-sieve co-reference resolution with knowledge. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Natural Language Learning,* Jeju Island, Korea, 12–14 July 2012, pages 1234–1244.

Don Tuggener. 2014. Coreference resolution evaluation for higher level applications. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics,* Gothenburg, Sweden, 26–30 April 2014, pages 231–235.

Olga Uryupina, Massimo Poesio, Claudio Giuliano, and Kateryna Tymoshenko. 2011. Disambiguation and filtering methods in using web knowledge for coreference resolution. In *Proceedings of the 24th International Florida Artificial Intelligence Research Society Conference,* Palm Beach, USA, 18–20 May 2011, pages 317–322.

Olga Uryupina. 2007. *Knowledge acquisition for coreference resolution*. Ph.D. thesis, Saarland University, Saarbrücken, Germany.

Olga Uryupina. 2008. Error analysis for learning-based coreference resolution. In *Proceedings of the 6th International Conference on Language Resources and Evaluation,* Marrakech, Morocco, 26 May – 1 June 2008, pages 1914–1919.

Renata Vieira and Massimo Poesio. 2000. An empirically-based system for processing definite descriptions. *Computational Linguistics*, 26(4):539–593.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Message Understanding Conference (MUC-6)*, pages 45–52, San Mateo, Cal. Morgan Kaufmann.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes release 5.0. LDC2013T19, Philadelphia, Penn.: Linguistic Data Consortium.