Natural Language Engineering **1** (1): 1–**??**. Printed in the United Kingdom
© Cambridge University Press

A Scaffolding Approach to Coreference Resolution Integrating Statistical and Rule-based Models

Heeyoung Lee[1], Mihai Surdeanu[2], and Dan Jurafsky[1]

[1]Stanford University, Stanford, California, USA
[2]University of Arizona, Tucson, Arizona, USA
heeyoung@cs.stanford.edu,
msurdeanu@email.arizona.edu,
jurafsky@stanford.edu

(*Received*)

Abstract

We describe a scaffolding approach to the task of coreference resolution that incrementally combines statistical classifiers, each designed for a particular mention type, with rule-based models (for sub-tasks well-matched to determinism). We motivate our design by an oracle-based analysis of errors in a rule-based coreference resolution system, showing that rule-based approaches are poorly suited to tasks that require a large lexical feature space, such as resolving pronominal and common-noun mentions. Our approach combines many advantages: it incrementally builds clusters integrating joint information about entities, uses rules for deterministic phenomena, and integrates rich lexical, syntactic, and semantic features with random forest classifiers well-suited to modeling the complex feature interactions that are known to characterize the coreference task. We demonstrate that all these decisions are important. The resulting system achieves 63.2 F1 on the CoNLL-2012 shared task dataset, outperforming the rule-based starting point by over 7 F1 points. Similarly, our system outperforms an equivalent sieve-based approach that relies on logistic regression classifiers instead of random forests by over 4 F1 points. Lastly, we show that by changing the coreference resolution system from relying on constituent-based syntax to using dependency syntax, which can be generated in linear time, we achieve a runtime speedup of 550% without considerable loss of accuracy.

1 Introduction

Coreference resolution—clustering expressions that refer to the same entity in a discourse—is an important component in most language understanding tasks, including text classification, information extraction, question answering, textual entailment, and summarization (Mitkov 2002; Steinberger et al. 2007; Gabbard et al. 2011; Mitkov et al. 2012; Kilicoglu et al. 2013).

Recent coreference systems have drawn on two distinct paradigms for the task. One is the standard supervised machine learning (ML) paradigm used throughout natural language processing. This has been applied to coreference by a number of classic systems (Connolly et al. 1994; McCarthy and Lehnert 1995; Kehler 1997; Soon et al. 2001; Ng and Cardie 2002; Rahman and Ng 2009) and more recently, enhanced with innovative ways to make use of broader information, such as the entity-mention model of Luo et al. (2004) and many kinds of global models (Mccallum and Wellner 2004; Daumé III and Marcu 2005; Denis and Baldridge 2007; Haghighi and Klein 2010). More recent machine learning advances have come from coreference-trees (Fernandes et al. 2012), ranking models (Durrett and Klein 2013), stacking models (Clark and Manning 2015), and neural models (Wiseman et al. 2015).

An earlier, classic line of work had focused on rule-based approaches to the special case of anaphora (Hobbs 1978; Lappin and Leass 1994), and early in the 20th century a number of scholars suggested that coreference might be a domain where such rule-based approaches might outperform machine learning (Stuckardt 2002; Zhou and Su 2004; Stuckardt 2005; Mitkov et al. 2007). Indeed, Haghighi and Klein (2009) showed that a coreference system based on deterministic syntactic/semantic rules could achieve the state of the art.

Drawing on these intuitions, Raghunathan et al. (2010) and Lee et al. (2011) and Lee et al. (2013) developed the *sieve* architecture, which relies on a sequence of hand-written rules (sieves), ordered from most to least precise. Like most modern architectures, the sieve architecture draws on rich linguistic features (Lee et al. 2013; Durrett and Klein 2013; Soon et al. 2001; Rahman and Ng 2009; Bengtson and Roth 2008) and is *entity-based* (Luo et al. 2004; Yang et al. 2008; Ng 2010; Clark and Manning 2015): decisions are made, not about mentions in the text, but about entities—clusters of mentions in the system's model of the world—allowing the system to reason about the properties of entities as a whole. The system's *precision ordering* allows it to first link high-confidence mention-pairs, and only later consider lower-confidence sources of information. The sieve-based approach was the top performer at the 2011 CoNLL challenge, and has become a component in state-of-the-art systems for many languages, including those that performed best in the 2012 CoNLL challenge in English (Fernandes et al. 2012) and Chinese (Chen and Ng 2012), and has been applied to various languages (Yuan et al. 2012; Zhang et al. 2012), domains (biomedical) (Gilbert and Riloff 2013; Jindal and Roth 2013), and tasks (like joint coreference/entity linking) (Hajishirzi et al. 2013).

Nonetheless, this rule-based sieve system, like all state of the art systems, is insufficient to address the coreference problems that hold back true language understanding.

Our first goal in this paper is to understand why. We therefore performed a detailed error analysis of the rule-based approach, identifying several important limitations such as poor performance on pronominal anaphora. Anaphora resolution depends on combining many small cues from the surrounding context, something that is hard for a rule-based system. Rule-based systems, in particular, do not deal well with rich lexical information, since it is difficult to write rules that correctly deal in advance with all possible context word situations. Dealing with rich lexical features is exactly where machine-learning based systems shine, since with sufficient training data and appropriate lexical templates they can learn the fine-grained lexical idiosyncracies that play such an important role in disambiguation in general.

Our results also show that errors pattern by *mention type*: linking content words depends on head-word semantics, pronoun errors are often caused by non-referential uses of *it* or *you*, while proper noun errors have to do with naming patterns in the real world.

Based on this error analysis, our second contribution is to explore an architecture for coreference resolution that maintains the scaffolding approach of the sieve architecture but integrates it with machine-learning-based models. In doing so, we draw on previous work that proposed to integrate the sieve architecture with machine learning (Denis and Baldridge 2008; Chen and Ng 2012; Ratinov and Roth 2012).

Our *hybrid architecture* consists of a series of sieves, like previous work, but incorporates both rule-based and statistical sieves trained by machine learning. Each statistical sieve is now designed around a particular mention type (common nouns, proper nouns, pronouns) and is based on random forests. Previous research has suggested that successful machine-learning coreference resolution is particularly dependent on feature conjunction: individual features don't give strong signals for coreference (Wiseman et al. 2015). Random forests naturally model large numbers of conjoined features. Further, our approach trains each sieve on the output of the previous sieves, which allows each sieve to be individually optimized on the appropriate examples in the context in which they appear. The resulting system is simple, intuitive, global, modular, and extensible.

We perform a number of experiments exploring the characteristics of this new hybrid architecture. Is the sieve-structured architecture better than simply throwing all the features into a single large classifier? Is it helpful to have some deterministic (non-machine learned) sieves? Does the ordering of sieves matter? Does the architecture facilitate the creation of useful features of entire entity clusters (rather than just features based on mentions)? Are random forests better suited for coreference resolution than linear classifiers? We answer these questions while showing that the new system works significantly better than previous systems.

Our third contribution is designed to address a performance problem with the sieve-based architecture: it is extremely slow. We analyze the run time of the end-to-end coreference resolution system and show that is dominated by the constituent-based syntactic parser, which accounts for 81% of the run time. To address this, we converted several of our components, such as mention detection and pattern-based sieves, to using dependency-based syntax, which can be generated in linear time (Chen and Manning 2014). We show that the entire coreference resolution system can be sped up by 5 times, at a loss of accuracy of only 1.1 CoNLL F1 points.

2 Error Analysis by Mention Type

We begin with an error analysis, inspired by and building on a number of previous analyses of errors in various coreference systems. For example Stoyanov et al. (2009) analyzed the performance of an early coreference resolution system by resolution classes. They found that proper nouns were easier to resolve than common nouns, that third person ungendered pronouns (*it* and *they*) were the most difficult to resolve, and that the accuracy of mention detection played a huge role in system performance.

The difficulty of pronouns has been confirmed by a number of previous analyses. Kummerfeld and Klein (2013) analyzed missing or extra mentions and mis-clustered entities, showing, e.g., the important role of pleonastic pronouns (*you*, *it*) in errors. Martschat and Strube (2014, 2015) propose a framework for comparing coreference errors, and evaluate such errors for entity-pair systems, finding that pronouns, and especially third-person genderless pronouns (*it, they*) are a particularly large sources of errors. Wiseman et al. (2015) also found that pleonastic pronoun mentions constitute a particularly large source of errors.

Other error analyses have pointed to large number of errors in common nouns. Recasens et al. (2013) showed that a particularly large source of error is caused by trying to link common noun mentions that do not have overlapping head words, e.g., *app* and *software*, a finding replicated also by Wiseman et al. (2015).

In this section we categorize errors by mention type, following these previous models, but focusing on a state-of-the-art sieve approach, and using an oracle method to investigate which words and situations are most problematic for each type of mention. An analysis by type has a particular advantage in our work because, unlike for most of the other systems previously analyzed, it aligns directly with our system components; e.g., many errors coming from resolving second person pronouns would suggest improvements in the pronoun-resolution component.

We investigate the following types of errors (examples in Table 1):

- *Mention Detection*: Errors in detecting correct mention spans or referentiality, including false positives that are *singletons* (whether non-referential or just mentioned once in the document), and false negatives (missed mentions) that are *mention recall* errors.

Type	`mention detection recall error`
Text	*“. . . **a security guard** at the intersection of the road towards Disney . . . ”*
Note	Correct mention in bold; the system selects the whole text due to incorrect syntax.
Type	`mention detection: singleton`
Text	*“. . . no cars can enter unless **they** have special permission . . . ”*
Note	The pronoun is generic and should not be considered as a mention.
Type	`proper - proper`
Text	*. . . people of the DPRK will be focused on inflicting the bitterest disasters upon **the United States of America**,” the statement read. . . . Reports recently surfaced that **the U.S.** was willing to consider bilateral talks . . .*
Note	Missed coreference link between *United States of America* and *the U.S.* (recall error)
Type	`common - common`
Text	*“Right now, there should be **seven main types of pipes buried underground**. They include those like you just mentioned, as well as **pipes like those for heating and communication**, among others. Truly, **these pipes** are closely linked to the lives of citizens.”*
Note	System incorrectly linked ***these pipes*** to the second, rather than the first, bold mention.
Type	`proper - common`
Text	***Barack Obama** yesterday outlined plans to close a loophole that lets companies ... avoid paying taxes on overseas profits. **The US President** used the Budget for 2016 . . .*
Note	Missed link between *Barack Obama* and *the US president*.
Type	`list - list`
Text	*“**The blood of goats and bulls and the ashes of a cow** were sprinkled on those who were no longer pure enough to enter the place of worship. **The blood and ashes** made them pure again – but only their bodies.”*
Note	Missed link between the two mentions in bold.
Type	`anytype - pronoun`
Text	*“Uh now look Lanny Davis **Sixty Minutes** gave Bill Clinton one hour last year ... He criticized **people**. He criticized Ken Starr. **They** didn’t have Ken Starr on to rebut.”*
Note	*They* incorrectly linked to *people* instead of *Sixty Minutes*.

Table 1. *Examples of coreference resolution errors, by type.*

- *Proper-Proper*: Errors when both mention and correct antecedent are proper nouns.
- *Common-Common*: Errors between two common-noun mentions.
- *Proper-Common*: Errors between a proper-noun antecedent and common-noun mention.
- *List-List*: Missing or incorrect links between mentions in lists or enumerations.
- *AnyType-Pronoun*: Pronoun resolution errors, with any type of antecedent. This includes errors in detecting generic pronouns.

Prior error analyses have generally drawn conclusions by studying the frequency of errors of different types. But in many coreference systems, and especially in entity-based systems like ours, errors influence each other; fixing one type of error could thus change the proportions of other errors. We therefore propose to directly study the impact of different errors on final system performance by using a partial oracle system. The partial oracle maintains the same sieves as the actual system, except the component being analyzed, which is replaced with perfect decisions.

We compare the best rule-based system (Lee et al. 2013), which serves as the starting point of our proposal, against these partial oracle systems. For example, to analyze proper-common errors, this oracle replaces system decisions with gold (perfect) decisions when resolving a common-noun mention with a proper-noun antecedent. This comparison lets us quantify "missed opportunities" from each component, and, consequently, the upper limit of coreference performance if the component had been perfect.[1]

System	CoNLL F1	diff
Rule-based system	56.08	-
Partial Oracle		
Mention Detection	74.58	18.5
→ Singleton Detection	68.35	12.27
→ Mention Recall	59.35	3.27
Proper-Proper	59.36	3.28
Proper-Common	57.77	1.69
Common-Common	62.36	6.28
List-List	56.19	0.11
AnyType-Pronoun	70.33	14.25
→ I	56.78	0.7
→ You	57.16	1.08
→ He	57.49	1.41
→ She	56.53	0.45
→ It	59.36	3.28
→ We	57.22	1.14
→ They	59.05	2.97

Table 2. *Error analysis of the rule-based system showing performance increase of the partial oracle. The score difference indicates the upper limit of improvement we can achieve by fixing the corresponding error type. The → indicates a subclass of the error type listed immediately above. The last rows containing pronoun types include all wordform variants with the same lemma, including accusative and reflexive pronouns. For example, the* I *row includes also* my, me *and* myself.

We perform this error analysis on the development partition of the CoNLL 2012 Shared Task

[1] This partial oracle also captures dependencies between components, e.g., fixing the common-common sieve might improve pronominal resolution. We ran a second version of an oracle experiment to isolate components, comparing perfect output against a second partial oracle, where only the sieve under investigation uses the actual resolution algorithm and all other sieves use oracle decisions. The results of this experiment led to the same importance ranking of components as the experiment we report here, and hence is not presented.

dataset (Pradhan et al. 2012). This data is a combination of newswire, broadcast news, weblog, telephone conversation, etc. The number of tokens in the development dataset is 160K (222 documents).

Table 2 lists the result of this experiment, from which we draw the following observations:

1) The most significant errors come from mention detection, followed by pronoun and common-common errors, consistent with previous error analyses. We discuss each below.

2) The mention detection oracle has a considerably higher score than the current system, consistent with the results of Stoyanov et al. (2009) and others. This is because the mention detection oracle solves two difficult tasks: (a) identifying correct mention spans (requiring perfect parses), and (b) determining referentiality of mentions. To understand the distribution of these errors, we conducted two further oracle experiments. In the *mention recall* experiment we added all mentions missed by the system (this fixes error type (a) above). In the *singleton detection* experiment we used gold data to mark a mention as singleton or not, since singleton mentions are not coreferent with any other mentions in the corresponding document (this fixes error type (b)). We found significantly more errors for error type (b), highlighting the difficulty of detecting referentiality; indeed, this is actually a good part of the coreference resolution problem itself, and previous error analyses have emphasized its importance (Wiseman et al. 2015). Mention recall errors are largely caused by parsing errors (Kummerfeld and Klein 2013). The fact that the performance improvement for this oracle is low indicates that parsing is already robust enough for this task. All in all, this clearly shows that mention detection is not a trivial task, and there is clear room for improvement.

3) As mentioned, pronominal resolution errors are the dominant error in coreference linking. The need for improvements in pronominal resolution has been reported for earlier systems and is a consistent theme in other analyses. The dominant pronominal error are the third person *it* and *they* mentions. The *it* errors are mainly due to failures of detecting pleonastic *it* pronouns, suggesting that the pleonastic pronoun detection in the deterministic system— a set of syntactic and lexical patterns drawing on earlier sets of patterns for pleonastic detection like Lappin and Leass (1994)— has insufficient accuracy. The *they* errors are caused by the ambiguity in gender and animacy of the pronoun *they*, which can corefer with most plural nouns, or (singular or plural) organizations.

4) Common-common errors are the third most dominant error type. This is due to the high frequency of the common-common coreferential links. But these errors are important also because these links likely require semantics for correct resolution, to understand, e.g., that *victim* and *casualty* are synonyms (Recasens et al. 2013).

5) While there are fewer errors involving proper-nouns than those involving common-nouns, consistent with all previous research, there are still enough that need to be addressed.

6) A few rule-based sieves—such as the speaker-based sieve, which focuses on first and second-person pronouns that appear in dialogue—work very well. For example, we generally solve first and second-person pronouns within 1 F1 point of the oracle system. This highlights that rule-based methods are not to be ignored: they perform very well when they need to manage few features and when the context can be modeled deterministically (e.g., speaker turns in dialogue).

In summary, the analysis suggests that while rule-based sieves perform well in certain situations, they have strong limitations when they need to manage a large lexical feature space, which is required for a better mention detection or pronominal resolution. This larger message hints towards a hybrid solution, which combines machine learning for scenarios that require many features with rule-based components for scenarios that may be complex but can be unambiguously modeled by a domain expert.

In the next section, we describe such a hybrid approach, implemented in the same sieve architecture which offers the right balance of simplicity (sieves can be independently constructed by different researchers), and power (incrementally building entity clusters allows the extraction of complex, global features patterns).

3 The Hybrid Architecture

The architecture of our model draws on the sieve architecture of Lee et al. (2013), with the goal of maintaining the elements that made that architecture successful while incorporating machine learning to address the problems identified in the error analysis above.

Like the Lee et al. (2013) model, we apply a series of resolution models sequentially, ordering the models so as to keep the precision in the earlier sieves as high as possible. while injecting the maximum amount of information on entities as early as possible. This strategy allows this model to incrementally gather reliable information about the entities in text and use it to guide future decisions. [2]

Our hybrid architecture consists of a series of seven sieves. Drawing on the insights of Chen and Ng (2012), we combined both rule-based and statistical sieves. Our final system combines two rule-based sieves and five statistical sieves trained by supervised machine learning, as shown in Figure 1. This architecture is simpler than the one proposed by Lee et al. (2013) (it has seven sieves, whereas Lee et al.'s has 10), yet, as we show later, it performs considerably better.

The five statistical sieves are each designed around a particular mention type: proper nouns, common nouns, mixes of the two, lists, and pronouns. Thus the `proper-proper` sieve is dedicated to resolving links from proper-noun mentions to proper-noun antecedents. The idea for this modular architecture comes from Ratinov and Roth (2012), who, drawing on the earlier work of Denis and Baldridge (2008), suggested that re-organizing sieves according to mention type allows them to differently weight features that may behave very differently for each type. Our results support Ratinov and Roth's finding; as we will see, lexical semantic features like word vectors are important for common nouns, while grammatical structure and morphological attributes are more important for pronouns.

Although our work thus draws heavily on the Ratinov and Roth (2012) statistical sieve models and other early work, we do explore some new architectural elements.

One is the classifiers themselves. Our approach is a hybrid one, combining rule-based and ML-based sieves, whereas the Ratinov and Roth model, for example, focuses on ML models alone. As we show below in our ablation study, combining rule- and ML-based components yields the best overall performance. We also used random forests for all the statistical classifiers, unlike earlier models which tend to use linear classifiers. Previous research has suggested that successful machine-learning coreference resolution is particularly dependent on feature conjunction: individual features don't give strong signals for coreference (Wiseman et al. 2015); this may explain the strong performance of deterministic systems, in which rule-writers can build rules that explicitly conjoin large numbers of features. Random forests, like all classifiers built on decision trees, very naturally model very larger numbers of conjoined features. Our

[2] As much as possible, we kept the sieve order from the deterministic model by aligning the sieves in the two approaches, e.g., the new "proper – proper" sieve approximately corresponds to the old "string match" one in (Lee et al. 2013). the goal of maximizing early information led us to change the original ordering in one situation: we placed the detection of appositives (Pass 2) before the "proper – proper" sieve due to the additional information it detects, despite its lower precision.

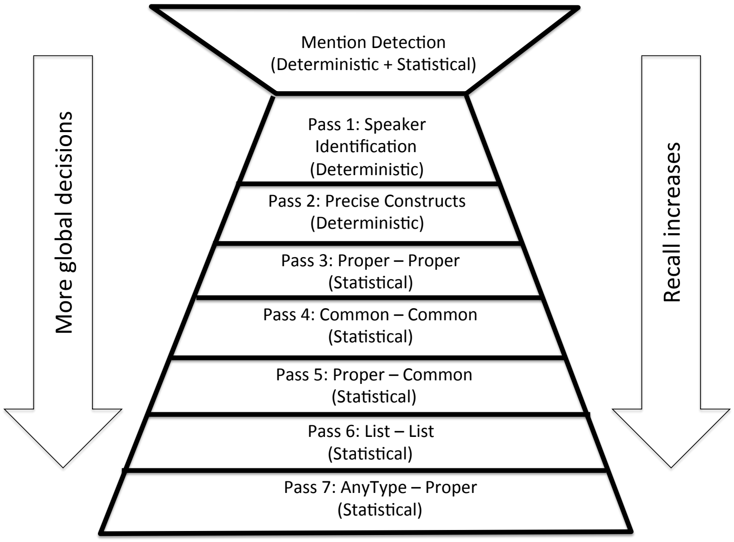

Mention Detection
(Deterministic + Statistical)
Pass 1: Speaker Identification (Deterministic)
Pass 2: Precise Constructs (Deterministic)
Pass 3: Proper – Proper (Statistical)
Pass 4: Common – Common (Statistical)
Pass 5: Proper – Common (Statistical)
Pass 6: List – List (Statistical)
Pass 7: AnyType – Proper (Statistical)
More global decisions
Recall increases

Fig. 1. Proposed hybrid architecture combining deterministic and statistical sieves.

experiments validate this observation: the approach that relies on random forests outperforms considerably the approach where the statistical classifiers are implemented using linear classifiers.

Our model incorporates a different decision model than that of Ratinov and Roth (2012). Similar to Lee et al. (2013), our model commits to decisions at the end of sieves, while Ratinov and Roth allow early decisions to be overridden by following sieves. Our model is thus less powerful, but has the advantage of less complicated bookkeeping. As we show, our approach achieves state-of-the-art performance despite the simpler model. Similar to Ratinov and Roth (2012) our hybrid model creates training data for a given sieve from the output of the previous sieves. This approach thus allows each sieve to be individually optimized on the appropriate examples in the context in which they appear in an actual disambiguation scenario. Similar to previous work, we found that this paradigm yields better performance.

Finally, we augmented the mention detection component of the new system, since we found that mention detection is one of the biggest sources of errors. We introduce a novel hybrid mention detector, which uses a rule-based component to propose mention candidates, and a ML component to disambiguate between multiple spans that share the same headword.

We note that our strategy that focuses on decomposing the task into simpler subtasks and using decision trees or deterministic models to address them differs considerably from recent trends in using neural networks for coreference resolution. For example, Wiseman et al. (2016) use recurrent neural networks to capture global representations of entity clusters directly from their mentions. Clark and Manning (2016) use a deep reinforcement learning approach to directly optimize a neural mention-ranking model for coreference resolution. Clark and Manning (2016) currently obtain the best performance on the English and Chinese portions of the CoNLL 2012 dataset. However, while neural-network approaches tend to perform well, they produce models that are hard to interpret, which increases their long-term maintenance cost (Sculley et al. 2014). Our approach mitigates this cost by producing a series of simpler models that are easier to interpret, as they rely either on deterministic approaches, or on decision trees that can be analyzed by human domain experts.

In the following subsections we describe our mention detection strategy, followed by details of our proposed sieves.

3.1 Mention Detection

As we have shown in §2, mention detection is the most significant source of errors in the rule-based system. Mention detection is complicated by the following two issues.

First, identifying singleton mentions improves the system's performance considerably, as shown in Table 2. However, in practice, few coreference resolution approaches identify singleton mentions ahead of time, because this is a complex task that is nearly identical to the complete coreference resolution problem, i.e., it requires the exploration of all possible antecedents, just not the selection of one. Following previous work, we also defer this decision to the subsequent sieves (see below).

Second, identifying correct mention spans is complicated by the imperfect syntax produced by existing parsers. For example, in the text *the car was stopped by [[a security guard] at the intersection of the road towards Disney]*, the parser incorrectly attached the PP *at the intersection of the road towards Disney* to the NP *a security guard*, instead of the verb. The mention detection component thus had difficulty deciding the end boundary for the mention headed by *guard* (see the two closing brackets). Choosing the wrong boundary is a considerable error, because it generates both a false positive (the longer mention) and a false negative (the shorter mention).

We propose a hybrid approach for mention detection that addresses the above issues with a combination of deterministic rules (driven by syntax) and statistical decisions to handle ambiguities. Our approach starts by marking most noun phrase constituents, named entities, and pronouns as *mention candidates*. We filter these candidates using a minimal set of exclusion rules: (a) following the OntoNotes annotation guidelines (Pradhan et al. 2007; BBN Technologies 2006), we discard adjectival forms of nations or nationality acronyms (e.g., *American, U.S., U.K.*); and (b) we remove the following stop words: *there, etc., ltd., 's, hmm, mm, ahem, um*.

The second component of our mention detection algorithm deploys a statistical classifier to identify correct mention boundaries when there are multiple mention candidates sharing the same headword, e.g., as in the *security guard* example above. In this situation, a binary classifier is deployed to select a single mention from all candidates sharing the same head word. Table 3 lists the features used by this classifier.

Our mention candidate list is by design over-inclusive. We let the following coreference resolution sieves handle the incorrect mentions in this candidate list by eventually marking them as singletons (i.e., by not linking them with any other mentions), which are removed from the output in a post-processing step.

3.2 Rule-based Sieves

Following mention detection, our approach deploys two rule-based sieves in succession, adapted from our previous work (Lee et al. 2013). These sieves capture complex patterns that yield deterministic resolution decisions. The **Speaker Identification** sieve links all first person pronouns to the speaker of the corresponding utterance. The **Precise Constructs** sieve identifies coreference relations that are driven

Named entity type
Words immediately preceding/following the mention
First/last word in the mention
Part-of-speech of the words in the above two feature groups
Whether this candidate is the larger NP or smaller NP
Whether another named entity with the same text exists in the same document
Whether or not the mention is the full span of a named entity

Table 3. *Features used by mention span classifier.*

by the following strong syntactic patterns: **appositive**, **predicate nominative**, **role appositive**, **relative pronoun**, **acronym**, and **demonym**[3].

These two sieves are deployed on all mention types (common nouns, proper nouns, pronouns), and are executed first because they are highly precise. Although they only capture a small subset of the existing coreference relations, the precise information these sieves extract about the entities can be exploited by the later statistical sieves.

3.3 Statistical Sieves

Motivated by the sieve architecture of Ratinov and Roth (2012) and our error analysis, we introduce five statistical sieves. Each error class from §2 corresponds to a distinct statistical sieve (or pass) in Figure 1. As Ratinov and Roth (2012) and Denis and Baldridge (2008) propose, designing sieves by mention class allows us to incorporate different knowledge into each sieve that is specifically useful for resolving mentions of that types. For example, linking common nouns requires synonym information that is computable from word vectors, whereas finding the antecedent of a pronoun requires information on attributes like animacy. We detail next the resolution algorithm, i.e., deciding if two given mentions belong to the same cluster or not at prediction time, followed by the training process for the statistical sieves, and the features used.

Resolution

Figure 2 illustrates the resolution algorithm that is used for all statistical sieves. When resolving a mention, (e.g., *his* in the example in the figure), the sieve attempts to link the mention to all possible antecedents within a certain sentence range. The sentence range may be different between sieves (d_i indicates the sentence-distance range for the i-th pass). Antecedents that are out of range are not considered. The antecedent candidate that yields a coreference link generated with the highest confidence (*the president* in the second sentence in Figure 2) is selected as the antecedent. We control for link over-generation by imposing a minimal confidence threshold, that, again, is specific to each sieve (t_i: merging threshold for i-th pass). If no confidence value is larger than t_i, then no coreferent antecedent is selected for the corresponding mention, i.e., the mention is currently considered a singleton (this state may change in the subsequent

[3] Demonyms are not annotated in OntoNotes. However, we found them to be useful as they help construct a richer entity context, which is used by the following sieves. The actual demonyms are discarded in a post-processing step that follows all sieves.

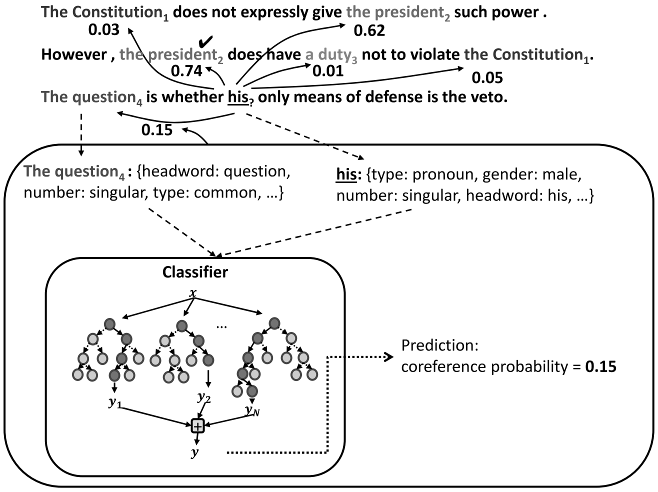

The Constitution$_1$ does not expressly give the president$_2$ such power .
0.03
0.62
However , the president$_2$ does have a duty$_3$ not to violate the Constitution$_1$.
0.74
0.01
0.05
The question$_4$ is whether his$_?$ only means of defense is the veto.
0.15
The question$_4$: {headword: question, number: singular, type: common, ...}
his: {type: pronoun, gender: male, number: singular, headword: his, ...}
Classifier
x
...
y_1
y_2
y_N
y
Prediction: coreference probability = 0.15

Fig. 2. Resolution in a statistical sieve, with mention subscript indices (and different colors) indicating current mention clusters. The example—the resolution of *his* in line 3— shows the classification of potential antecedent *The question*. The system ultimately chooses the higher confidence answer (*the president*).

sieves). All d_i and t_i hyper-parameters are tuned on the OntoNotes development set. Importantly, we use both mention-pair and cluster-pair features for all statistical sieves (see §3.4).

Training

Each statistical sieve trains on the output produced by the previous sequence of sieves on the training data. Thus, at each pass, a sieve is exposed to a list of mentions and a set of (partial) clusters that group the mentions according to decisions of the previous sieves. From this data, for each mention, we extract at most one positive (coreferent) and possibly multiple negative (non-coreferent) training data points consisting of the pair of the corresponding mention and one antecedent, where the antecedent must appear within the sentence range for this sieve.

For the coreference resolution task, the number of negative training examples is considerably larger than the number of positive examples (the former is quadratic in the number of mentions in a document, whereas the latter is linear). To mitigate excessive training times and to guarantee that the task fits in memory, we implemented a subsampling process for negative examples. We deploy this subsampling component for the sieves where the number of negative examples is excessive (currently the *proper–proper* and *common–common* sieves). The subsampling process is the following:

(1) We start by training a classifier for the relevant sieve using all positive examples gathered for all mentions in the training dataset, and a random subsample of 20% of the negative examples.

(2) We then inspect the classifier confidence values (estimated probabilities) for *all* the negative examples, and keep only the top 20% most-ambiguous negative examples, i.e., whose confidence values for belonging to the positive class is high. We train the final classifier using these more informative negative examples and all positive examples.

This method is conceptually similar to methods that favor training examples near the class boundaries (Burges 1998).

All classifiers are implemented using random forests (Breiman 2001). The only hyper-parameters tuned were the percentage of features to be used for each split point, and the number of individual decision trees[4].

Example Document

[Russian Foreign Minister Igor Ivanov]$_1$ congratulated Kostunica$_2$ on [[his]$_2$ election victory]$_3$
He$_1$ also gave him$_?$ a letter from Russian President Vladimir Putin.

E/ M	Example value	**Feature**
M	1	Sentence distance between the two mentions
M	2	Mention distance between the two mentions based on antecedent ordering. The value is 2 because is the second antecedent considered.
E	1	Minimum sentence distance between any two mentions from each cluster. In the example, the cluster$_2$: $\{$*Kostunica, his*$\}$ and cluster$_?$: $\{$*him*$\}$ have sentence distances 1 and 1, with the minimum 1
M	-	Clause distance – # of clauses (“*S**” constituents in parse tree) between two mentions when they occur in the same sentence
-	No	Document type: is this document conversational text?
-	“cnn”	Document source: the example is from CNN
M	“narrator”	Who is the speaker of the anaphor mention?
M	1 / 1	the (antecedent / anaphor) mention length
M	No	Is length of antecedent (2 in example) longer than the length of the anaphor mention (1 in example)?
E	2 / 1	How many mentions are currently in the (antecedent / anaphor)’s cluster?
M	“dobj” / “iobj” / ”dobj-iobj”	What is the role (subject or object) of the (antecedent / anaphor), and their combination?
M	“antecedent starts with definite article = No”	The (antecedent / anaphor) starts with (definite / indefinite) article (*’the’* / *’a’*, *’an’*)
M	“mention is an indefinite pronoun = No”	The (antecedent / anaphor) (is/starts with) an indefinite pronoun (*anybody*, *something*, etc.)
M	“antecedent is a reflexive pronoun = No”	The (antecedent / anaphor) is a reflexive pronoun (*herself*, etc.)
M	No	The headword is the last word of the anaphor mention
M	No	The anaphor has a *Wh-* or *that-* phrase after the headword
M	“mention doesn’t have indefinite article nor post phrase”	The combination of definiteness of the anaphor mention and post-headword phrase above
E	“P-Pr” / “Pr”	The list of mention types for the mentions in the (antecedent / anaphor) cluster: (C)ommon, (P)roper, (Pr)onoun
M	“S-VP-NP-NNP” / “S-VP-NP-PRP”	The path in the parse tree from the root to the (antecedent / anaphor)
E	5 / 7	The sentence # that includes the first mention in the (antecedent / anaphor) cluster
M	“antecedent is bare plural = No”	Whether the (antecedent / anaphor) is (bare plural / quantifier / partitive / % / demonym / ’etc.’)
M	No	Whether the later mention is pleonastic *it*
M	No	Whether the antecedent or anaphor have negation modifier: e.g., *no books*
M	singular, male, animate, unk, person / singular, male, animate, third, other	The number, gender, animacy, person, named entity type attributes of (antecedent / anaphor)
M	singular-singular, animate-animate, male-male, unk-third, person-other	The combination of the attributes of antecedent and anaphor
E	singular, male, animate, unk, person / singular, male, animate, third, other	The number, gender, animacy, person, named entity type attributes of the (antecedent / anaphor) cluster
M, E	all agree	Whether each attribute agrees
E	No	Any mention in the anaphor cluster has i-within-i relation with any mention from the antecedent cluster
M	No	Antecedent is the speaker detected for the anaphor
M	Yes (narrator)	The speaker of antecedent and anaphor are the same
M	No	Antecedent and anaphor are subject and object of the same sentence
E	No	Any mention from the antecedent cluster and any mention from the anaphor cluster have subject-object relation
M	No	Person attributes disagree but speakers are identical
E	No	The strings of any two mentions from the antecedent and anaphor clusters are identical. Similar to ExactStringMatch in Lee et al. (2013)
E	No	The strings of any two mentions from the antecedent and anaphor clusters, after removing any *Wh-* or *that-* phrases that follow the headwords, are the same. Similar to RelaxedExactStringMatch in Lee et al. (2013)
M	No	Headwords are identical

continued…

[4] We used the default value, unlimited-depth, for the maximum depth of the decision trees.

Example Document
[Russian Foreign Minister Igor Ivanov]$_1$ congratulated Kostunica$_2$ on [[his]$_2$ election victory]$_3$ He$_1$ also gave him$_?$ a letter from Russian President Vladimir Putin.

E/ M	Example value	**Feature**
M	No	Headwords are identical, and both are proper mentions
E	No	Any two mentions from each cluster have the same headword
E	No	Any two mention from each cluster have the same headword, and both are proper mentions
M	No	The anaphor mention has a proper noun which is not in antecedent
E	No	Both clusters have at least one proper mention
M	No	Anaphor and antecedent mentions contain different location named entities (LOC)
E	No	Each cluster has at least one incompatible modifier with mentions in the other cluster
E	No	Any mention in the anaphor cluster is an acronym of another mention from the antecedent cluster
M	No	The anaphor and antecedent are in the (appositive / predicate nominative / role appositive) relation. Similar to PreciseConstruct in Lee et al. (2013)
M	No	The anaphor mention contains a number, which is not found in the antecedent mention
E	No	Any word in a mention span from the anaphor cluster appears in the antecedent cluster
M	-	The number of elements if the anaphor is an enumeration.
M	3rd person pronoun	The type of the pronoun (masculine pronoun, ’you’, possessive, etc.)
M	VBD	The preceding and following word or part-of-speech tag of *you know*
M	No	The anaphor and antecedent mentions are both (’I’ / ’you’)
M	No	The anaphor and antecedent are an ’I’ and its speaker relation
M	No	The anaphor mention is a reflexive pronoun and the antecedent is its subject
M	No	The anaphor and antecedent have different speakers and they are: ’I’, ’you’, ’we’
M	No	The (antecedent / anaphor) is ’you’ in ’you know’
M	“mHeadPOS-PRP” / “mHeadword-him” etc	The (part-of-speech/lexical) feature of (headword / first / last / preceding / following word) of (antecedent / anaphor mention)
E	“him” / “him”	The (headwords / words) in the anaphor cluster which are not a (headword / word) of any mention in the antecedent cluster
M	0.089	The Euclidean distance between the two headwords in their corresponding word vectors
M	0.089 / 0.089 / 0.249 / 0.301	The distance between the two first / last / preceding / following words in their word vectors
M	0.089	The distance between the two aggregate vectors (the sum of all word vectors in the mention)
M	0.089	The average distance between any two word vectors from each mention

Table 4: Features (using entity (E) or mention (M) information) that compare an anaphor mention with a candidate antecedent; showing in the second column sample features extracted between *Kostunica* and *him* in the given example document.

3.4 Features

We list in Table 4 the features used by all statistical sieves. As the table shows, features may exploit information that is local to the mention (M), or that is global, from the currently available cluster of mentions pointing to the same entity (E). For example, one feature aggregates gender information from all the mentions in the antecedent’s cluster.

We incorporated all the features used by the rules in Lee et al.’s system (2013), plus part-of-speech and lexical features from Durrett and Klein’s system (2013) and word-embedding features from Mikolov et al. (2013) to model semantic similarity. The vectors were generated with the skip-gram model using 200 dimensions over the English Gigaword dataset (LDC2011T07). The features that use these vectors include the Euclidean distance between the vector representations of headwords, first and last words in mention/antecedent, immediately preceding and following words, and average distances of all words in both mentions.

Importantly, each statistical sieve uses a selected subset of this feature space, only retaining features that occur at least 20 times in the training data and have a pointwise mutual information (between feature and corresponding class) over 0.0001 (chosen by coarse tuning on the development dataset). As we show in the next section, the sieves adapt to their corresponding tasks by choosing very different features.

3.5 Coreference Resolution Using Dependency-based Syntax

The proposed coreference system (similar to most others) uses a constituent parser for several purposes. These include extracting noun phrases for mention detection, finding headwords that are required by various sieves, and finding syntactic relations such as apposition, which are used in the precise syntactic constructions sieve.

Our analysis of the run time of the end-to-end coreference resolution system revealed that the syntactic parser dominates the execution time: on average, 81% of the run time is spent on constituent parsing. This is because the constituent-based syntactic parser used in our system (Klein and Manning 2003) relies on probabilistic context free grammars (PCFG), which are applied using a chart-based dynamic programming algorithm with a runtime complexity of $O(N^3)$, where N is the number of words per sentence. By contrast, the actual coreference resolution algorithm has a theoretical runtime complexity of $O(M^2)$, where M is the number of mentions in a document, because a mention could be linked to any other mention in the document. Furthermore, in practice this is $O(M)$, because we constrain the search for antecedents to a small number of preceding sentences.

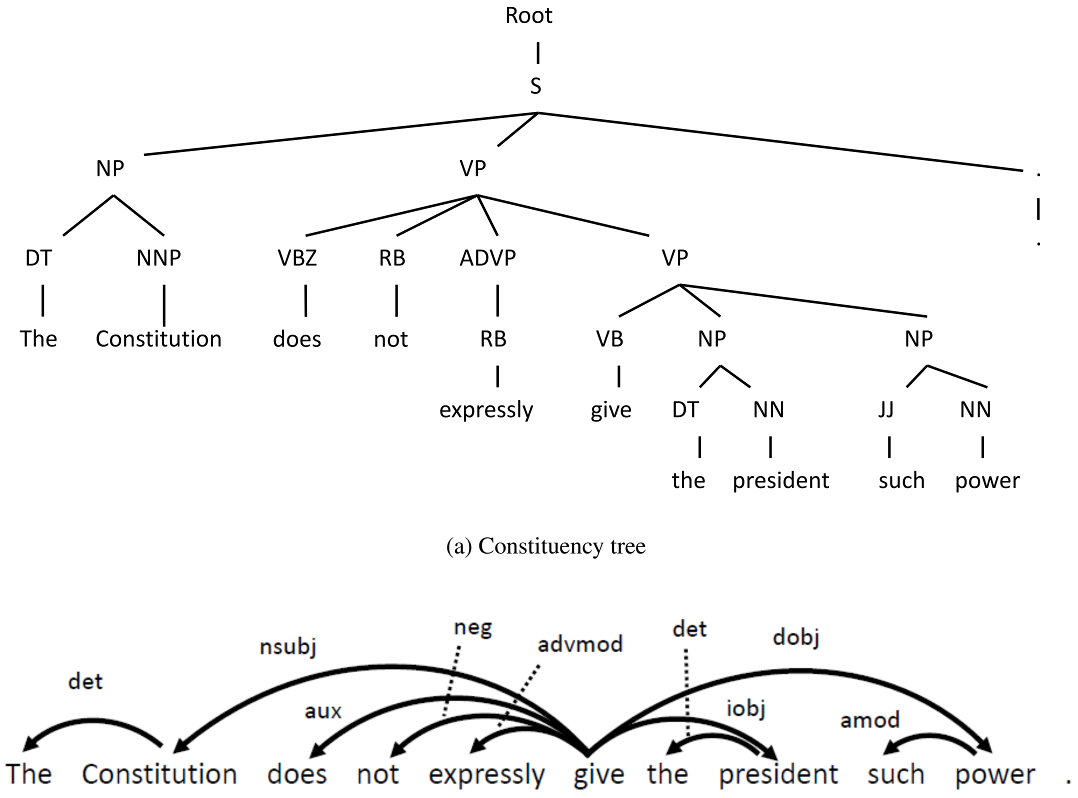
While there are more efficient algorithms for constituent parsing (Petrov and Klein 2007; Roark and Hollingshead 2008; Yi et al. 2011), in general dependency parsing is faster (Cer et al. 2010). Unlike the constituent-based representation, the dependency-based representation is flat, i.e., each node (or word) is connected to other nodes by a directed grammatical relation such as nominal subject. An example contrasting the two syntactic representations is shown in Figure 3. Because of this simpler representation, dependency parsers using the shift-reduce algorithm achieve state-of-the-art performance at a run time linear in the sentence length (Chen and Manning 2014). Previous work has shown that, despite their simplicity, these dependency parsers perform similarly with constituent parsers for more complex language processing tasks such as discourse parsing (Surdeanu et al. 2015).

3.5.1 Using a dependency parser instead of a constituency parser

While the above analysis of run times is good motivation for switching to dependency-based syntax, this implies several nontrivial changes throughout the coreference resolution system. We discuss them below. We chose the Stanford dependency representation (de Marneffe and Manning 2008) as our new syntactic representation. We generate it in two different ways. For the runtime analysis, we generate Stanford dependencies directly by replacing our original constituent parser (Klein and Manning 2003) with the dependency parser of Chen and Manning (2014). For the qualitative comparison against other systems that evaluated on the CoNLL 2012 shared task data (Pradhan et al. 2012), we start with the constituent parse trees that are included in the dataset and were used by all previous work, and convert them to Stanford dependencies using software included in the CoreNLP toolkit (Manning et al. 2014).

Mention detection: One of the main reason the constituent syntax is widely used for coreference resolution is due to mention detection. Our coreference resolution system (similarly to most others) considers noun phrases, named entities, and pronouns as mentions to resolve. Out of these, noun phrases require parsing to be identified. As shown in Figure 3, noun phrases (NP) can be trivially extracted from constituent trees, but they are not explicitly marked in dependency trees.

To identify NPs in dependency trees, we implemented the following algorithm. For all nodes whose part-of-speech in the dependency tree is `N*`, `PRP*`, or `DT*`, we take the subtree of the node (i.e., all nodes syntactically dominated by this node) as a mention, as long as the node does not serve as a determiner (*det*) or noun modifier (*nn*) for another node. If the subtree contains a copular relation (*cop*), we



(a) Constituency tree

(b) Dependency tree, using Stanford dependencies

Fig. 3. Parse trees for the sentence *”The Constitution does not expressly give the president such power.”*

remove all words dominated by this dependency up to the governor of the copular relation. For example, in Figure 3, there are three nouns *Constitution, president*, and *power*. The corresponding noun phrases are created simply by taking the subtree governed by these three nouns in the dependency tree, yielding the following three NPs: *The Constitution*, *the president*, and *such power*.

We show an additional, more complex example in Figure 4 (showing the constituency tree above the text, and the dependency tree below). In this tree, there are two `DT`s: *This* and *the*, and three nouns (`N*`): *reason*, *price*, and *instability*. Below we show the system decisions for each of these mentions (and other non-mentions):

1) **Mention**: *This* — its part-of-speech is `DT`, and it is not in a determiner relation with another word.
2) **Not Mention**: *the* — its part-of-speech is `DT`, but it serves as modifier in a *determiner (det)* relation with another word (*reason*).
3) **Mention**: *price instability* — this is a phrase covered by the subtree of the node *instability*, which has an `N*` part-of-speech.
4) **Not Mention**: *price* — this has a `N*` part-of-speech, but it is in a *noun modifier (nn)* relation with another word (*instability*).
5) **Not Mention**: *This is the reason for price instability*—this is a phrase covered by a subtree of the

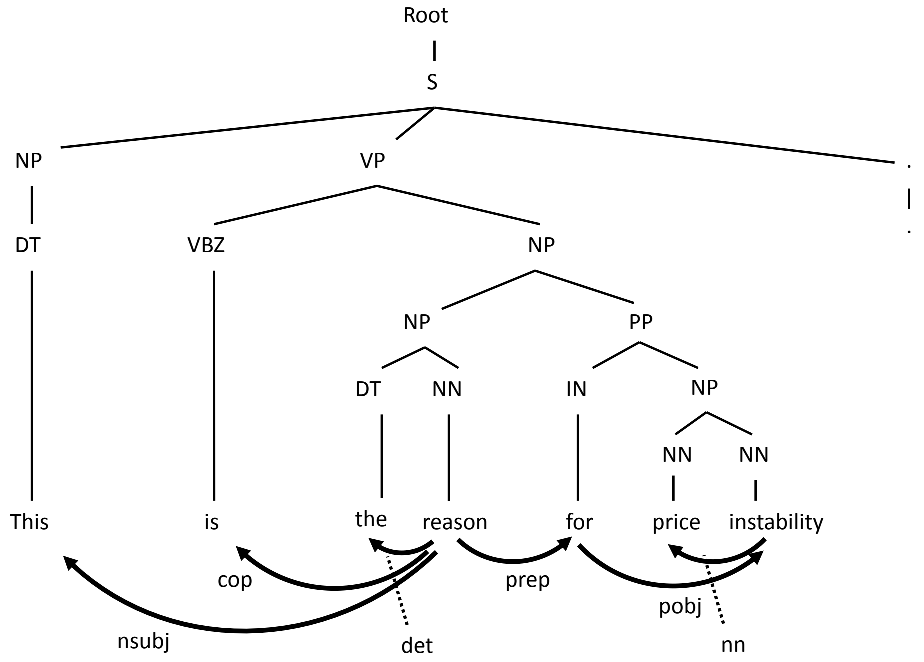


Fig. 4. An example of mention detection based on constituency tree or dependency tree

node *reason*, which has an N* part-of-speech. However, one of its child has the copular relation *is - cop - reason*.
6) **Mention**: *the reason for price instability* — this is a phrase covered by the subtree of the node *reason*, after removing all words up to the word in the copular relation (i.e., *is*).

Headword detection: Headword information is crucial in coreference resolution. For example, headword agreement is the most important feature in the resolution of proper and common noun anaphors.

For constituency syntax, there are a number of widely-used linguistic heuristics for discovering headwords of constituent phrases (Collins 1999). For dependency syntax, by contrast, the extraction is much simpler, because dependency relations are directed from governor (or headword) to modifier. Exploiting this information, the headword of a sequence becomes the only word whose own head is outside of the sequence. For example, in Figure 3, *The* cannot be the headword of the phrase *The Constitution*, because its own headword (*Constitution*) is included in the sequence. On the other hand, the governor of *Constitution* (*give*) is outside of the phrase. Thus *Constitution* becomes the headword for the entire sequence. Below are the headwords of all other mentions identified for the text in Figure 3 by this algorithm:

headword(*the president*) = *president*
headword(*such power*) = *power*
headword(*This*) = *This*
headword(*price instability*) = *instability*

headword(*the reason for price instability*) = *reason*

Syntactic relation detection: Particular syntactic structures like appositions or copular relations give very strong clues for coreference (Lee et al. 2013). When we use a constituency tree for coreference, these structures are discovered using regular expression patterns over constituent trees (Levy and Andrew 2006). In contrast, the dependency tree labels these relations explicitly as *cop* and *appos* dependencies, so they are trivial to extract. For example, in Figure 4, *reason* and *This* are in a copular relation. As a result, the precise patterns sieve links the mentions *the reason for price instability* and *This*.

4 Evaluation

We evaluate our system on the English portion of the CoNLL 2012 Shared Task dataset (Pradhan et al. 2012). This dataset is a combination of newswire, broadcast news, weblog, telephone conversation, etc., and contains 1.6M words and 2384 documents. Similar to the shared task, we calculate five coreference metrics: MUC (Vilain et al. 1995), B^3 (Bagga and Baldwin 1998), $CEAF_m$, $CEAF_e$ (Luo 2005), and CoNLL F1 (average of MUC, B^3, and $CEAF_e$), using the CoNLL 2012 scorer version 8.

4.1 Results

Table 5 compares the performance of our system under various configurations against previous work.[5] The table includes three configurations of the random forest classifiers (for 100 and 1000 decision trees trained on training set, and 1000 trees trained on train+dev dataset), three ablation experiments, where we remove important components of our system, and two artificial configurations that analyze the impact of the scaffolding architecture. The hyper parameters for all sieves were coarsely tuned to optimize the CoNLL F1 score on the development set. In particular, we used a threshold of 0.2 for merging two mention clusters for the pronoun sieve, and a threshold of 0.3 for all the other sieves (these apply over the merging probability as produced by the RF). For the limit in sentence distance between two mentions we used: 5 sentences for the pronoun sieve, no limit for the proper – proper sieve, and 15 for all other sieves. The feature subset size that the RF uses is: 50 features for the proper – proper sieve, and 30 for all other sieves. All sieves used unpruned trees.

4.2 Discussion

The table shows that our best configuration (rows 9 or 10) gives comparable performance with a current state-of-the-art system (Wiseman et al. 2015), and outperforms by nearly 8 CoNLL F1 points the rule-based system of Lee et al. (2013), which served as the starting point of this work. This demonstrates that it is possible to achieve roughly state-of-the-art performance while maintaining the modularity and most of the simplicity of the original sieve architecture.

The experiments summarized in Table 5 allow us to answer several additional questions:

Are the rule-based sieves still important? The ablation experiment in row 11 shows that the rule-based

[5] For all results in the table we used version 8 of the CoNLL scorer, which fixes several bugs from the version used during the shared task. For this reason, these results are different from the ones reported in Pradhan et al. (2012).

			MUC			B^3			CEAF$_m$			CEAF$_e$			
	System	# trees	R	P	F1	R	P	F1	R	P	F1	R	P	F1	CoNLL F1
1	Lee et al. (2013)		65.08	62.41	63.72	50.23	54.08	52.08	58.45	53.95	56.11	54.01	44.27	48.65	54.82
2	Fernandes et al. (2012)		65.83	75.91	70.51	51.55	65.19	57.58	57.48	65.93	61.42	50.82	57.28	53.86	60.65
3	Durrett and Klein (2013)		66.58	74.94	70.51	53.2	64.56	58.33	59.19	66.23	62.51	52.9	58.06	55.36	61.4
4	Björkelund & Kuhn (2014)		67.46	74.3	70.72	54.96	62.71	58.58	60.33	66.92	63.45	52.27	59.4	55.61	61.63
5	Durrett and Klein (2014)		69.91	72.61	71.24	56.43	61.18	58.71	-	-	-	54.23	56.17	55.18	61.71
6	Clark and Manning (2015)		69.38	76.12	72.59	56.01	65.64	60.44	-	-	-	52.98	59.44	56.02	63.02
7	Wiseman et al. (2015)		69.31	76.23	72.60	55.83	66.07	60.52	-	-	-	54.88	59.41	57.05	63.39
8	This work	100	68.55	76.03	72.10	55.56	65.63	60.18	60.99	68.82	64.66	51.60	61.83	56.25	62.84
9		1000	68.82	76.02	72.24	55.75	65.69	60.32	61.29	68.89	64.87	51.94	62.02	56.53	63.03
10		1000 train+dev	68.75	76.4	72.37	55.82	65.94	60.46	61.15	69.15	64.91	51.99	62.5	56.76	63.2
11	− Rule-based sieves	100	66.68	76.59	71.29	51.52	67.10	58.28	58.09	67.32	62.37	50.66	60.55	55.17	61.58
12	− MD classifier	100	67.21	77.49	71.99	53.95	67.26	59.88	59.70	70.00	64.44	50.31	62.53	55.76	62.54
13	− Entity features	100	69.14	74.72	71.82	56.43	62.66	59.38	60.33	65.68	62.89	51.61	57.60	54.44	61.88
14	Single pass	100	71.87	69.08	70.42	60.03	57.16	58.49	63.92	62.61	63.24	54.09	56.60	55.31	61.41
15	Reverse ordering	100	68.39	75.79	71.90	55.32	65.22	59.86	60.71	68.23	64.25	51.76	61.06	56.03	62.60
16	Dependency tree	100	66.70	76.42	71.23	53.65	65.80	59.10	59.22	69.22	63.83	49.30	61.83	54.86	61.73
17	Logistic regression	-	65.46	73.31	69.16	50.65	62.49	55.95	56.92	63.79	60.16	48.73	54.76	51.57	58.89

Table 5. *Performance of our approach compared to previous work. Similarly to previous work, we used the syntactic parse trees and named entities included in the CoNLL corpus. "# trees" is the number of individual decision trees used by the random forest classifiers. Row 10 shows score of the model trained on both train and development dataset. Rows 11 – 13 show three ablation experiments: "− Rule-based sieves": removing the first two rule-based sieves, "− MD classifier": removing the statistical component of the mention detection module described in Section 3.1, and "− Entity features": removing the entity level features (E) in Table 4. Rows 14, 15, and 16 show three different configurations. "Single pass": using a single statistical model with all features from Table 4 and no rule-based sieves, and "Reverse ordering": all sieves, both rule-based and statistical, but called in reverse order to the one in Figure 1. "Dependency tree": using dependency syntax instead of constituency. The dependency trees are generated by converting the constituency tree provided with the dataset to Stanford dependencies using CoreNLP software. All hyper parameters for rows 11 – 16 are identical to the configuration in row 8. We repeated the comparison experiments (rows 8 and 11 – 16) 10 times and averaged the scores. A non-parametric permutation test indicates row 8 is statistically significantly better than rows 11 – 16 ($p < 0.01$). Row 17 shows the performance of the best sieve architecture (row 10) but replacing each RF classifier with a linear, logistic regression classifier. We used the CoNLL 2012 scorer version 8 for all reported scores.*

sieves contribute 1.26 F1 points to the overall performance (row 8). This comes on top of statistical sieves that contain features that capture the same information as the rules (e.g., our "narrator" feature in Table 4 captures if the current mention refers to the speaker, the same information used by the speaker detection sieve). In the statistical sieves, these features are drowned out by many other features, which is avoided by the deterministic process in the rule-based sieves. This result illustrates the benefit of hybrid architectures.

How important is the scaffolding architecture? The scaffolding architecture improves two aspects of the singular classifier approach: (a) it captures (some of) the jointness of the task by incrementally constructing mention clusters, which are then available for the extraction of global features ((E) in Table 4), and (b) it reduces the error rate by keeping the antecedent search space small in each sieve (e.g., only

System	CoNLL F1 with original CoNLL preprocessing	Preprocessing time	Coreference resolution time	Total time	Tokens / Sec	Memory requirement	CoNLL F1 with Stanford CoreNLP preprocessing
Constituent	62.72	749	11	760	223	2.77G	60.5
Dependency	61.62	132	5	137	1238	2.70G	60.62

Table 6. *Overall runtimes, i.e., starting from raw text, on the entire CoNLL shared task 2012 test dataset, in seconds. We ran the two configurations on a single machine with two Intel Xeon CPUs E5-2660 @ 2.20GHz (16 cores and 32 hyperthreading) and 128GB of RAM. The first CoNLL F1 reported (first column) is provided for reference from the previous table: it was measured when the system used the CoNLL-provided preprocessing (parser, NER, part-of-speech tagging). For the rest of the columns, we used Stanford CoreNLP version 3.5.1 for all preprocessing. We used the CoNLL scorer version 8 for all scores.*

proper nouns in the proper-proper sieve). We verified the importance of these two issues in rows 13 and 14, respectively. Row 13 shows that global (entity-based) features contribute 0.96 F1 points, whereas solving all mentions jointly in a single classifier drops performance by 1.43 F1 points (as mentioned in the table caption, rows 13 and 14 are compared against row 8). This demonstrates that, while joint information is indeed useful, the biggest benefit of the sieve architecture comes from the scaffolding of components. We cannot be sure, however, what portion, if any, of this improvement is due to the fact that having separate classifiers means we have more hyper parameters to tune (see Section 4.1). Row 17 shows the performance of the linear classifier system with the same sieve architecture and features.

What is the impact of sieve ordering? Lee et al. (2013) show the importance of precise-first ordering of sieves. To verify this for our hybrid approach, in row 15 we show the performance of an experiment where all seven sieves are called in reverse order (from *pronoun* to *speaker id*). The performance drop of 0.24 F1 points indicates that sieve ordering is not crucial, even though it is statistically significant.

What is the impact of the statistical component in mention detection? The ablation experiment in row 12 removes the statistical component of mention detection, which disambiguates between multiple mentions with the same head word, showing this component accounts for a statistically-significant improvement of 0.3 F1 points. This is not a considerable contribution, but, to the best of our knowledge, our work is the first to show that this type of mention disambiguation is beneficial to the overall task.

Is there a performance penalty when switching to dependency syntax? When we switched from the constituent to the dependency parser, we measured a 1.11 CoNLL F1 points decrease (row 16 compared to row 8). The performance difference is caused mainly by a weaker mention detection component and less robust syntactic coreference patterns (e.g., appositives) when relying on dependency syntax. On the other hand, Table 6 shows that the runtime of this configuration decreased by more than fivefold compared to the original system that used the Stanford parser (Klein and Manning 2003). All in all, we believe that, for most real-world applications, a 1 F1 point drop is an acceptable compromise for a speedup of fivefold.

What is the impact of the forest size? The table 5 shows minimal performance differences with different numbers of trees (rows 8 and 9), indicating that it is possible to build good performing models with a relatively small memory footprint. Table 6 shows that we can run this end-to-end coreference model with 3 GB of RAM.

Are random forests better than linear classifiers? We chose random forests because of their ability to handle complex feature interactions; Wiseman et al. (2015) and others have shown that individual features aren't sufficiently informative for coreference. To validate this observation, we compared our best RF-based sieve model (row 10) against a similar model, where each classifier was replaced with a linear, logistic regression (LR) one (row 17). This comparison show that, indeed, the LR-based model performs over 4 F1 points worse than the corresponding RF-based model. This confirms that modeling the non-linearity of the feature space for coreference resolution is important.

What are the random forests learning? Analyzing the trees learned by the RF classifiers, we found that the model was indeed making use of very large conjunctions of features. Figure 5 shows decision paths (from the root of one decision tree to the leaf) from the common-common sieve and the pronoun sieve. The depth of these decisions are 174 and 121, meaning that the tree considers the interaction of a vast number of features, far greater than easily possible in most other types of classifiers. Features in the first path include the similarity of head words, definiteness, mention length, textual overlap, distance, semantic similarity (from word vectors), etc. We observed similarly complex paths in other sieves, focusing on different features. For example, the pronominal resolution sieve focuses on animacy and number agreement and context words.

the mention string from the beginning up to the headword is the same (F)
→ headwords are the same (T)
→ mention distance is less than 14.5 (i.e., there are fewer than 14.5 mentions in between) (T)
→ the part-of-speech of the first word of the mention is “DT” (T)
→ mentions have incompatible modifiers (F)
→ the mention starts with an indefinite (F)
→ antecedent starts with a definite (T)
→ aggregate difference in word vectors is larger than 0.17 (T)
→ antecedent length is smaller than 7.5 (T)
→ mention distance is larger than 6.5 (T)
→ words in mention are included in words in antecedents (T)
→ ...
→ mention distance is less than 14.5 (T)
→ mention length is less than 2.5 (T)
→ minimum sentence distance is larger than 2.5 (T)
→ **MENTIONS ARE COREFERENT**

both mention and antecedent are `I` (F)
→ both are animate (F)
→ the cluster of mention has inanimate attribute (T)
→ antecedent is animate, mention is inanimate (F)
→ both mention and antecedent are `it` (F)
→ mention is followed by ”NNS” (F)
→ antecedent is followed by ”VBD” (F)
→ number attribute agree (T)
→ antecedent cluster size is larger than 1.5 (T)
→ antecedent is followed by ”.” (F)
→ sentence distance is less than 1.5 (T)
→ ...
→ antecedent is a definite mention (T)
→ both mentions are not named entity (T)
→ **MENTIONS ARE COREFERENT**

Fig. 5. Part of the decision path in the common-common sieve for linking mention *The industry* with antecedent *The oil industry's* and the part of decision path in the pronominal sieve for linking the pronoun *its* with the antecedent *The company*; (T) and (F) represent deciding True or False.

Sieve	Feature	Wt
Mention Detection	included in other NP	8.42
	includes other NP	1.61
	following word is ’of’	1.57
	preceding POS is ’NNP’	1.51
	last POS is ’NNP’	0.78
	first word is ’the’	0.53
	following word is ’.’	0.52
	first POS is ’PRP’	0.45
Proper-Proper	non-pronominal headword agreement among mentions in each cluster	0.49
	mention distance	0.24
	following POS of mention is ’NNP’	0.22
	mention string is exactly same	0.20
	relaxed headwords agreement	0.13
	antecedent cluster size	0.12
	sentence distance	0.11
	minimum sentence distance	0.11
Common-Common	non-pronominal headword agreement among mentions in each cluster	0.67
	words in antecedent cluster includes all words in mention cluster	0.17
	mention distance	0.15
	word vector average distance between words from each cluster	0.06
	mention length	0.05
	mention string is exactly same	0.04
	sentence distance	0.04
	modifiers in mention and antecedent are incompatible	0.04
Proper-Common	non-pronominal headword agreement among mentions in each cluster	0.028
	mention string is exactly same	0.024
	words in antecedent cluster includes all words in mention cluster	0.021
	mention length	0.011
	antecedent cluster size	0.008
	mention cluster size	0.006
	mention distance	0.005
	minimum sentence distance	0.005
List-List	word vector aggregate distance	0.087
	word vector headword distance	0.06
	word vector last word distance	0.043
	word vector average distance	0.04
	word vector first word distance	0.006
	first appearance of this mention?	0.007
	document source	0.007
	both mentions are animate	0.007
AnyType-Pronoun	all attribute agreement	0.86
	i-within-i	0.853
	mention distance	0.59
	antecedent cluster size	0.511
	minimum sentence distance	0.392
	sentence distance	0.352
	person disagreement	0.158
	document source	0.124

Table 7. 8 *most important features of each sieve, sorted in descending order of the permutation feature importance (Wt) assigned by the model. The permutation importance of a feature is calculated by mean decrease in accuracy when the feature is randomly permuted (Breiman 2001).*

Table 7 shows the top 8 features for each statistical sieve. The table shows that the feature space covers lexical, syntactic, and semantic features, and, crucially, that the important features differ between sieves. For example while proper and common nouns both rely on headword agreement among all the mentions in the cluster, common nouns (and lists) further rely on word vector features. Pronominal anaphora resolution

System	CoNLL F1	diff
Hybrid System	63.29	-
Partial Oracle		
Mention Detection	73.9	10.61
→ Singleton Detection	69.95	6.66
→ Mention Recall	64.86	1.57
Proper-Proper	66.57	3.28
Proper-Common	65.24	1.95
Common-Common	69.82	6.53
List-List	63.34	0.05
AnyType-Pronoun	74.61	11.32
→ I	63.77	0.48
→ You	64.8	1.51
→ He	64.5	1.21
→ She	63.61	0.32
→ It	66.14	2.85
→ We	64.41	1.12
→ They	65.83	2.54

Table 8. *Performance increase of the partial oracle system on the development dataset. The score difference is the max improvement achievable by fixing the corresponding error type. The → indicates a subclass of the preceding error type. Similar to Table 2, the last rows containing pronoun types include all wordform variants with the same lemma.*

relies most on agreement of attributes like person and number. This is a further argument for a modular approach for coreference resolution.

4.3 Error Analysis

To understand the improvements that our hybrid system has over the baseline rule-based system of Lee et al. (2013), we repeated the partial-oracle analysis described in Section 2 for the proposed hybrid system.

The results, summarized in Table 8, show improvements for both mention detection and anaphora resolution. The performance gap in mention detection between our hybrid system and the oracle mention detection system (10.61) is smaller than that of the rule-based system (18.5). This demonstrates that our new mention detection algorithm, together with better coreference resolution, significantly reduces the performance drop due to incorrect mention detection.

The gap between our system and the oracle pronominal resolution system also decreased (from 14.25 to 11.32), showing that our system successfully captures and combines the various weak signals for pronoun resolution. For example, the oracle error for *it* pronouns decreases from 3.28 to 2.85. Our conjecture is that since our model employs as a binary feature the deterministic pleonastic detection from Lee et al. (2013), some of the improvements seen come from feature weighting and composition operations that include this feature. Nonetheless, the high remaining error rate for this class of pronouns highlights the need for more sophisticated models of pleonastic pronoun detection (Boyd et al. 2005; Müller 2006; Wiseman et al. 2015; Wiseman et al. 2016).

5 Conclusion

We describe a hybrid approach for coreference resolution, combining rule-based and machine-learning components in a sieve architecture. Our approach maintains the modular architecture and precise-first ordering of our prior deterministic sieve (Lee et al. 2013), but changes almost everything else.

First, drawing on the insights of Denis and Baldridge (2008) and Ratinov and Roth (2012), each sieve is designed around a particular mention/antecedent type (e.g., common nouns, proper nouns, or pronouns). Second, driven by thorough error analysis, we pragmatically choose the best approach for each sieve, interleaving rule-based approaches (e.g., to handle precise syntactic constructs such as predicate nominatives) with statistical models (e.g., to handle the larger context necessary for pronominal resolution). We used the same hybrid strategy for mention detection, using patterns to propose mention candidates and a statistical model to disambiguate between overlapping mentions.

Driven by the intuition that feature conjunctions are important for coreference, our statistical sieves were built using random forests, allowing them to capture the complex feature interactions and rich lexical features that characterize the coreference task, while maintaining the interpretability of the resulting model. We empirically demonstrated that modeling the non-linear interactions between coreference resolution features is indeed important: the approach based on random forests classifiers outperforms by more than 4 F1 points the equivalent system where the classifiers were implemented using logistic regression.

Our approach is simple, intuitive, easy to train, modular, and extensible, yet captures the main advantages of joint learning and entity-based modeling. We show that each aspect of our system contributes to its performance: separating decision into sieves, ordering the sieves by precision, mixing rule-based and statistical sieves, and using hybrid mention detection.

Finally, we demonstrate that the slow performance of the sieve algorithm is caused mainly by constituent parsing. We show how to convert the sieve algorithm to use dependency rather than constituency parses, resulting in a system that is 5 times faster with only a slight loss in F1 score. This speedup is critical for real world applications. In general, we believe that the faster hybrid system for coreference can play an even more central role in a wide variety of NLU applications in the future. To encourage adoption, we are releasing our system as open-source software. The code is available at: `https://github.com/heeyounglee/hcoref`.

References

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

BBN Technologies. 2006. Coreference Guidelines for English OntoNotes – Version 6.0.

Eric Bengtson and Dan Roth. 2008. Understanding the value of features for coreference resolution. In *Proceedings of EMNLP 2008*, pages 294–303, Honolulu, Hawaii.

Anders Björkelund and Jonas Kuhn. 2014. Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In *Proceedings of ACL 2014*, pages 47–57, Baltimore, Maryland.

Adriane Boyd, Whitney Gegg-Harrison, and Donna Byron. 2005. Identifying non-referential *it*: a machine learning approach incorporating linguistically motivated features. In *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in NLP*, pages 40–47, Ann Arbor, Michigan.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Christopher J.C. Burges. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167.

Daniel M. Cer, Marie-Catherine De Marneffe, Daniel Jurafsky, and Christopher D Manning. 2010. Parsing to Stanford dependencies: Trade-offs between speed and accuracy. In *Proceedings of LREC 2010*, pages 1628–1632, Valletta, Malta.

Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP 2014*, pages 740–750, Doha, Qatar.

Chen Chen and Vincent Ng. 2012. Combining the best of two worlds: A hybrid approach to multilingual coreference resolution. In *Proceedings of EMNLP-CoNLL 2012*, pages 56–63, Jeju Island, Korea.

Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. In *Proceedings of ACL*, pages 1405–1415, Beijing, China.

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

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Dennis Connolly, John D. Burger, and David S. Day. 1994. A machine learning approach to anaphoric reference. In *Proceedings of the International Conference on New Methods in Language Processing (NeMLaP-1)*, pages 255–261, UMIST, Manchester.

Hal Daumé III and Daniel Marcu. 2005. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *HLT-EMNLP 2005*, pages 97–104, Vancouver, B.C., Canada.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Proceedings of COLING Workshop on Cross-framework and Cross-domain Parser Evaluation*, pages 1–8, Manchester, UK.

Pascal Denis and Jason Baldridge. 2007. Joint determination of anaphoricity and coreference resolution using integer programming. In *Proceedings of NAACL-HLT 2007*, pages 236–243, Rochester, NY.

Pascal Denis and Jason Baldridge. 2008. Specialized models and ranking for coreference resolution. In *Proceedings of EMNLP 2008*, pages 660–669, Honolulu, HI.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of EMNLP-2013*, Seattle, Washington.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *TACL*, 2:477–490.

Eraldo Rezende Fernandes, Cícero Nogueira dos Santos, and Ruy Luiz Milidiú. 2012. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *EMNLP-CoNLL*, pages 41–48, Jeju, Republic of Korea.

Ryan Gabbard, Marjorie Freedman, and Ralph Weischedel. 2011. Coreference for learning to extract relations: Yes Virginia, coreference matters. In *ACL 2011*, pages 288–293, Portland, Oregon.

Nathan Gilbert and Ellen Riloff. 2013. Domain-specific coreference resolution with lexicalized features. In *Proceedings of ACL 2013*, pages 81–86, Sofia, Bulgaria.

Aria Haghighi and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of EMNLP 2009*, pages 1152–1161, Suntec, Singapore.

Aria Haghighi and Dan Klein. 2010. Coreference resolution in a modular, entity-centered model. In *Proceedings of HLT-NAACL 2010*, pages 385–393, Los Angeles, CA.

Hannaneh Hajishirzi, Leila Zilles, Daniel S. Weld, and Luke Zettlemoyer. 2013. Joint coreference resolution and named-entity linking with multi-pass sieves. In *Proceedings of EMNLP 2013*, Seattle, Washington.

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

Prateek Jindal and Dan Roth. 2013. Using domain knowledge and domain-inspired discourse model for coreference resolution for clinical narratives. *Journal of the American Medical Informatics Association (JAMIA)*, 20(2):356–362, 3.

Andrew Kehler. 1997. Probabilistic coreference in information extraction. In *Proceedings of EMNLP 1997*, pages 163–173, Providence, Rhode Island.

Halil Kilicoglu, Marcelo Fiszman, and Dina Demner-Fushman. 2013. Interpreting consumer health questions: The role of anaphora and ellipsis. In *Proceedings of the 2013 Workshop on Biomedical Natural Language Processing*, pages 54–62, Sofia, Bulgaria.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL 2003*, pages 423–430, Sapporo, Japan.

Jonathan K. Kummerfeld and Dan Klein. 2013. Error-driven analysis of challenges in coreference resolution. In *Proceedings of EMNLP 2013*, pages 265–277, Seattle, Washington.

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

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In *Proceedings of CoNLL 2011: Shared Task*, pages 28–34, Portland, Oregon.

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.

Roger Levy and Galen Andrew. 2006. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of LREC 2006*, pages 2231–2234, Genoa, Italy.

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the Bell tree. In *Proceedings of ACL 2004*, pages 21–26, Barcelona.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of HLT-EMNLP 2005*, pages 25–32, Vancouver, B.C., Canada.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of ACL 2014*, pages 55–60, Baltimore, Maryland.

Andrew Mccallum and Ben Wellner. 2004. Conditional models of identity uncertainty with application to noun coreference. In *Proceedings of NIPS 2004*.

Joseph F. McCarthy and Wendy G. Lehnert. 1995. Using decision trees for coreference resolution. In *Proceedings of IJCAI 1995*, pages 1050–1055, Montréal.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. NIPS Foundation.

Ruslan Mitkov, Richard Evans, Constantin Orasan, Le An Ha, and Viktor Pekar. 2007. Anaphora resolution: To what extent does it help NLP applications? In A. Branco, editor, *Proceedings of DAARC 2007*, volume 4410 of *LNAI*, pages 179–190, Berlin / Heidelberg. Springer-Verlag.

Ruslan Mitkov, Richard Evans, Constantin Orăsan, Iustin Dornescu, and Miguel Rios. 2012. Coreference resolution: To what extent does it help nlp applications? In *International Conference on Text, Speech and Dialogue*, pages 16–27. Springer.

Ruslan Mitkov. 2002. *Anaphora Resolution*. Longman, London.

Christoph Müller. 2006. Automatic detection of nonreferential it in spoken multi-party dialog. In *Proceedings of EACL 2006*, pages 49–56, Trento, Italy.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of ACL 2002*, pages 104–111, Philadelphia.

Vincent Ng. 2010. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of ACL*, pages 1396–1411, Uppsala, Sweden.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of HLT-NAACL 2007*, pages 404–411, Rochester, New York.

Sameer S. Pradhan, Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2007. OntoNotes: A unified relational semantic representation. In *Proceedings of ICSC*, pages 517–526.

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 Sixteenth Conference on Computational Natural Language Learning (CoNLL 2012)*, Jeju, Korea.

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 (EMNLP)*, pages 492–501, Cambridge, Massachusetts.

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

Lev Ratinov and Dan Roth. 2012. Learning-based multi-sieve co-reference resolution with knowledge. In *Proceedings of EMNLP-CoNLL 2012*, pages 1234–1244, Jeju Island, Korea.

Marta Recasens, Matthew Can, and Dan Jurafsky. 2013. Same referent, different words: Unsupervised mining of opaque coreferent mentions. In *Proceedings of NAACL 2013*, pages 897–906, Atlanta, Georgia.

Brian Roark and Kristy Hollingshead. 2008. Classifying chart cells for quadratic complexity context-free inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING)*.

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, and Michael Young. 2014. Machine learning: The high interest credit card of technical debt. In *SE4ML: Software Engineering for Machine Learning (NIPS 2014 Workshop)*, Montreal, Canada.

Wee M. Soon, Hwee T. Ng, and Daniel C. Y. Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Josef Steinberger, Massimo Poesio, Mijail Alexandrov Kabadjov, and Karel Jezek. 2007. Two uses of anaphora resolution in summarization. *Information Processing and Management*, 43(6):1663–1680.

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 ACL-IJCNLP 2009*, pages 656–664, Suntec, Singapore.

Roland Stuckardt. 2002. Machine-learning-based vs. manually designed approaches to anaphor resolution: the best of two worlds. In *In: Proc. 4th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC2002), University of Lisbon, Sept. 2002*, pages 211–216.

Roland Stuckardt. 2005. A machine learning approach to preference strategies for anaphor resolution. In A. Branco, A. McEnery, and R. Mitkov, editors, *Anaphora Processing: Linguistic, Cognitive and Computational Modeling*, pages 47–72. John Benjamins.

Mihai Surdeanu, Thomas Hicks, and Marco A. Valenzuela-Escárcega. 2015. Two practical rhetorical structure theory parsers. In *Proceedings of NAACL-HLT 2015*, Denver, Colorado, USA.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of MUC-6*, pages 45–52.

Sam Wiseman, Alexander M. Rush, Stuart M. Shieber, and Jason Weston. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Proceedings of ACL-IJCNLP 2015*, pages 1416–1426, Beijing, China.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Proceedings of NAACL 2016*, pages 994–1004, San Diego, CA.

Xiaofeng Yang, Jian Su, Jun Lang, Chew L. Tan, Ting Liu, and Sheng Li. 2008. An entity-mention model for coreference resolution with inductive logic programming. In *Proceedings of ACL-HLT 2008*, pages 843–851, Columbus, Ohio.

Youngmin Yi, Chao-Yue Lai, Slav Petrov, and Kurt Keutzer. 2011. Efficient parallel CKY parsing on GPUs. In *Proceedings of the 2011 Conference on Parsing Technologies*, pages 175–185, Dublin, Ireland, October.

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 *Proceedings of EMNLP-CoNLL 2012*, pages 76–82, Jeju, Republic of Korea.

Xiaotian Zhang, Chunyang Wu, and Hai Zhao. 2012. Chinese coreference resolution via ordered filtering. In *Proceedings of EMNLP-CoNLL 2012*, CoNLL '12, pages 95–99, Jeju, Republic of Korea.

Guodong Zhou and Jian Su. 2004. A High-Performance Coreference Resolution System using a Constraint-based Multi-Agent Strategy. In *Proceedings of COLING 2004*, pages 522–529, Geneva, Switzerland.