When Annotation Schemes Change Rules Help: A Configurable Approach to Coreference Resolution beyond OntoNotes

Amir Zeldes and Shuo Zhang
Department of Linguistics, Georgetown University
{amir.zeldes,ssz6}@georgetown.edu

Abstract

This paper approaches the challenge of adapting coreference resolution to different coreference phenomena and mention-border definitions when there is no access to large training data in the desired target scheme. We take a configurable, rule-based approach centered on dependency syntax input, which we test by examining coreference types not covered in benchmark corpora such as OntoNotes. These include cataphora, compound modifier coreference, generic anaphors, predicate markables, i-within-i, and metonymy. We test our system, called xrenner, using different configurations on two very different datasets: Wall Street Journal material from OntoNotes and four types Wiki data from the GUM corpus. Our system compares favorably with two leading rule based and stochastic approaches in handling the different annotation formats.

1 Introduction

Previous work (Rahman & Ng 2011, Durrett & Klein 2013) has suggested that a trainable coreference resolution approach can outperform rule-based approaches (e.g. Lee et al. 2013) because of its ability to model similar constraints in a lexicalized way that more closely matches training data. However, in many cases the amount of training data required for such approaches is large: if the phenomenon that we wish to include is not annotated in the data, we can only use a trainable system after considerable annotation work to adjust the training set to include it. Permutations of what to include or exclude and how to model each phenomenon, can compound such problems further.[1]

Rule-based approaches (Haghighi & Klein 2009, Lee et al. 2013), by contrast, can more easily add new behaviors, but have been described as “difficult to interpret or modify” (Durrett & Klein 2013: 1971). Although they can achieve results competitive with trainable systems, the hard-wired aspects of rule-based systems are problematic if we wish to adapt to different annotation schemes, languages, and target domains.

The current paper approaches the challenge of different target schemes with a system called *xrenner*: an externally configurable reference and non-named entity recognizer. By using a large number of highly configurable mechanisms and rules in easily modifiable text files, with almost no hard-wired language- or domain-specific knowledge, we are able to adapt our system to include or exclude a variety of less standard coreference phenomena, including cataphora, generic indefinite anaphors, compound modifier nominals, predicate markables, clause-nested markables (i-within-i) and metonymy. We test our system on two datasets with very different schemes: Wall Street Journal data from OntoNotes (Hovy et al. 2006), which does not include the above cases, and a small test corpus, GUM (Zeldes 2016), which captures these phenomena and more.

[1] These limitations also apply to low resource languages (e.g. Sikdar et al. 2013 for Bengali) and domain adaptation (e.g. biomedical data, Apostolova et al. 2012, Zhao & Ng 2014), where large tailored training data is unavailable.

Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2016), co-located with NAACL 2016, pages 92–101,
San Diego, California, June 16, 2016. ©2016 Association for Computational Linguistics

2 The phenomena

Because of its size and quality, OntoNotes has become an established training and test set for coreference resolution. However, the OntoNotes annotation scheme (BBN Technologies 2007) does not cover several potentially useful and interesting phenomena, such as cataphora, predicatives, indefinite generic coreference, common noun compound modifiers, metonymy, and nested coreference.[2] These are illustrated below with cases from OntoNotes, which are not actually annotated in the corpus:

(1) **Cataphora:** [*it*]*'s certainly true* [*the rout began immediately after the UAL trading halt*]

(2) **Predicative:** [*He*] *is* [*an avid fan of a proposition on next week's ballot*]

(3) **Generic:** [*Program trading*] *is "a racket,"…* [*program trading*] *creates deviant swings*

(4) **Compound modifiers:** *small investors seem to be adapting to greater* [*stock market*] *volatility … Glenn Britta … says he is "factoring"* [*the market's*] *volatility "into investment decisions."*

(5) **Metonymy:** *a strict interpretation of a policy requires* **[***The U.S.*] *to notify foreign dictators of certain coup plots …* [*Washington*] *rejected the bid …*

(6) **Nesting:** *He has in tow* [*his prescient girlfriend, whose sassy retorts mark* [*her*] *…*][3]

It is certainly debatable whether or not the above phenomena should be treated as cases of coreference, or relegated to syntax (cataphora can be described as a purely syntactic phenomenon, i.e. as expletives) or semantics (predicatives may be considered complex predicates, not constituting markables for annotation). There are nevertheless cases in which we would be interested in each of these, and different corpora and language traditions have handled them differently, with direct consequences for systems trained on such corpora and their evaluation (see Recasens & Hovy 2010). While the above phenomena are not annotated in OntoNotes[4], many coreference resolution systems for English do in fact use, for example, predicative markables internally to facilitate coreference matching, even if the evaluation and output are set to delete them (cf. Lee et al. 2013).

The interest in diverse types of coreference relations has led to projects annotating them (notably ARRAU, Poesio & Artstein 2008), but as of yet, there is no training data source on the scale of OntoNotes that includes all of them. Because of this, the ability to configure a system to include or exclude such relations seems desirable: if we cannot assemble enough data to output these based on training alone, we need to use rules. But the different combinations of rules we might need depending on the target scheme require a flexible, configurable approach. In the next section we will outline our system, which relegates a wide range of coreference criteria to external configuration files, and includes treatments of the above phenomena.

3 A Configurable Framework

3.1 Core System Configuration

The xrenner system is an open source end-to-end entity recognition and coreference resolution system written in Python.[5] The input to the resolution components is dependency syntax data in the tabular CoNLL format, which can be produced by a parser; in experiments below we will use the Stanford Parser (Chen & Manning 2014) with Collapsed Typed Dependencies (CTDs). The decision to use dependencies is related to the configurability that it allows: we can define the desired mention

[2] Another phenomenon worth mentioning is bridging, which we will not deal with here, e.g. *Mexico's President Salinas said* [*the country*]*'s recession had ended and* [*the economy*] *was growing again.* (economy = the country's economy).

[3] An anonymous reviewer has noted that for some (non-singleton) mentions, nested pronouns are annotated, e.g. in document a2e_0020: "[*The American administration who planned carefully for this event through experts in media and public relations, and* [*its*] *tools*]". Under singleton mention, however, the nested pronoun is left unresolved, cf. another example: "*an elusive sheep with a star on its back*" (singleton notwithstanding nesting, not annotated in OntoNotes).

[4] A partial exception is metonymy, which is sometimes annotated as regular coreference, e.g. "*Mrs. Hills lauded* [*South Korea*] *…* [*Seoul*] *also has instituted …*" and sometimes ignored, as in the example above. Often, similar lexemes can appear as non-coreferent, making metonymy detection very challenging: e.g. *Japan … Tokyo's brat pack* (referring to a group of authors in Tokyo, not Japan in general).

[5] See `https://github.com/amir-zeldes/xrenner` for source code and `https://corpling.uis.georgetown.edu/xrenner/` for a live demo.

borders using dependency function chains in which certain dependencies are set to ‘break’ the chain. For example, if we include the relative clause CTD label, *rcmod*, (cf. de Marneffe & Manning 2013), we can easily decide to exclude these and ‘de-nest’ cases like (6). Such settings are configured for each resolution model in text files as regular expressions. The OntoNotes markable definition does not exclude relative clauses and is configured as:

```
non_link_func=/nsubj|cop|dep|punct|ap
pos|mark|discourse|parataxis|neg/
```

This means that mention borders propagate across all dependency functions not matching this expression. The annotation scheme used in the GUM corpus (see Section 4.1) has mentions excluding relative clauses, which can easily be modeled by adding *rcmod* to the setting above. Editing such settings can therefore radically alter the output of the system with very little effort.

The main configuration currently has over 70 settings of this type, including:

- Function labels for subject, coordination, etc., used in subsequent rules (see Section 3.4)
- Functions and tokens signaling modification (to collect a list of modifiers for each head, for coreference matching, see Section 3.4)
- Dependent strings and tags assigning a definiteness status after mention detection (articles, possessives), as well as numerals assigning cardinality (e.g. a modifier *three* maps to cardinality |3| for English)
- Dependent tags or functions required to match in coreference (e.g. possessives, or proper modifiers)
- POS tags which may serve as mention heads, including tags only admissible with certain functions (e.g. numbers, tagged *CD*, only as core arguments, not modifiers)
- Morphological agreement classes to assign to certain POS tags (e.g. map *NNS* to ‘plural’ agreement), as well as classes to assign by default, or in particular to coordinate markables (e.g. map coordinate mentions to ‘plural’, recognized via inclusion of the coordination function)
- Language specific settings such as whether person names must be capitalized, whether to attempt acronym matching, how questions and quotation are marked (relevant for direct speech recognition), and more
- Optional stemming for recognizing coreference between definite markables with no antecedent and a verb of the same stem (e.g. [*required*] … [*the requirement*])
- Postprocessing settings such as deleting certain function markables from the output (e.g. noun modifiers or copula predicates, based on CTD labels such as *nn* and *cop*)
- Surrounding appositions with joint markables (OntoNotes style), or deleting coordinations with no distinct mentions

The latter pair of settings, for example, can alter coreference chain output substantially, since according to OntoNotes, (7) would require two separate entity IDs (‘apposition wrapping’), whereas in (8) the coordination NP requires no coreference at all (no coordinate markables without aggregate mention):

(7) [[*five other countries*]$_i$ -- [*China, Thailand, India, Brazil and Mexico* --]$_i$]$_j$ … [*those countries*]$_j$

(8) [*The U.S.*] *and* [*Japan*] … [*The U.S.*] *and* [*Japan*]

3.2 Mention detection and entity resolution

The system performs its own entity type resolution and does not rely on existing NER software. Candidate mentions are recognized via dependency subgraphs as defined by eligible POS heads and linkable dependency functions. Based on the presence of certain modifiers defined in the configuration, properties such as definiteness and cardinality are assigned during mention detection.

Candidate entities are matched against multiple lexical resources, which contain major entity types such as PERSON, LOCATION, TIME, ORGANIZATION, ABSTRACT and more, as well as subclasses, such as POLITICIAN (subclass of PERSON), COUNTRY (subclass of PLACE), COMPANY (subclass of ORGANIZATION) etc. Agreement information can also be provided optionally (e.g. most likely gender for each proper name, or complete grammatical gender information for languages other than English; see below for sources). The model we will evaluate below distinguishes 11 major entity types and 54 subclasses, but the types and number of entity classes and subclasses are not constrained by the system. Instead they are derived directly from the lexicon files, allowing for different scenarios based on the lexical data available for the language and scheme being modeled. The system uses several lexicon files, which it consults in order:

- Entity list – full text of multi-token entities
- Entity heads – single token entity heads
- Entity modifiers – mapping of modifiers which identify the entity type, such as *President X* (PERSON), *X Inc*. (COMPANY), etc.
- Proper name list – first and last names for recognizing persons not in the entity list
- Paraphrase list – for non-identical lexical matching (i.e. 'is-a' relations, such as *company* → *firm*)
- Antonym list – gives incompatible modifiers that counter-indicate coreference (e.g. *the good news ≠ the bad news*)

The sources of the data for the English model evaluated below are summarized in Table 1.

Data	*Sources*
Proper names	DBPedia (Auer et al. 2007)
Geo-names	DBPedia (Auer et al. 2007)
Common nouns	GUM, OntoNotes
Is-a list	GUM, OntoNotes, PPDB (Gan-tikevitch et al. 2013)
Antonyms	OntoNotes, WordNet (Fellbaum 1998)
Named entities	GUM, OntoNotes, Freebase (Bollacker et al. 2008)

Table 1: Lexical resources used for the English model evaluated below.

Beyond explicit lexical resources such as DBPedia (Auer et al. 2007), WordNet (Fellbaum 1998) and Freebase (Bollacker et al. 2008), which provide lists of companies, politicians, animals and more, we use entity type labels from the training sections of OntoNotes and GUM. The system also benefits greatly from the Penn Paraphrase Database data (PPDB, Ganitkevitch et al. 2013), which contains a large amount of entries found to be equivalent translations in parallel corpora. These complement coreference information from GUM and OntoNotes, and help win some of the 'uphill battle' of contextually synonymous lexical NPs (cf. Durrett & Klein 2013). Entity entries from all sources, including entity head lexemes and modifiers (e.g. *Mrs.*), can be specified as 'atomic', in which case the mentions they identify may not contain nested mentions. This will be crucial for ruling out spurious compound modifier coreference below.

The is-a table is also the basis for our handling of metonymy, by including e.g. entries for capitals mapped to their countries (the assumption is that such metonymy usually occurs after the country has been explicitly mentioned, so we do not include the opposite direction). Multiple entries are allowed for each key in the lexicon, so a *bank* can be a PLACE (river bank) and an ORGANIZATION (financial institution). Disambiguation and resolution of unknown entity strings is carried out based on a mapping of dependencies to entity types taken from GUM and OntoNotes training data (e.g. a subject of *barked* is typically of the class ANIMAL).

When this data is missing, the longest suffix match in the lexicon is used (e.g. *vitrification* is classed as EVENT if the longest suffix match with the lexicon is *-ification*, and most items with this suffix are events). As a result, we have a chance of catching metonymy by ruling between alternate entries for an entity as e.g. a country, if it is the dependent of a head that more typically governs a country (for example, a *prep_against* dependent of the word *embargo*). In essence, this means we treat metonymy as a word sense disambiguation problem.

All nominals are assigned an entity type, so that entity type resolution is not restricted to proper name entities, and all pronoun entities are initially guessed via dependency information of the type above, within their respective agreement classes.

3.3 Post-Editing Dependencies

Input dependency trees can be manipulated by a Python module called DepEdit[6], which takes rules identifying relevant tokens via features and graph relationships (token distance or parentage sub-

[6] See `https://corpling.uis.georgetown.edu/depedit/`

graphs), and reassigns new functions or subgraphs based on the configuration. Rules take the form:

$$
\begin{aligned}
&R = <Tok_{i..j}, Rel_{k..l}, Act_{m..n}> \\
&\quad Tok = \{f_1..f_k\} \mid f \in \{\text{text,lemma,func,head}\} \\
&\quad Rel = <Tok, op, Tok> \mid op \in \{>, . , .\text{n}, .\text{n,m}\} \\
&\quad Act = \{f_i \rightarrow g_i\}
\end{aligned}
$$

Such that a token definition is matched based on the features f_i, designating the token text, lemma, head or dependency function (usually as a regular expression), and relationships are binary constraints on pairs of tokens, via an operator indicating the head-dependency relation (>) or adjacency (.), potentially within *n-m* tokens. Each action Act_i is a mapping of some feature value to a new value (e.g. changing POS or function), including the 'head' feature, which allows rewiring of dependency trees.

Table 2 shows two such rules, one for handling a certain cataphoric construction, and another for handling age appositions. The first rule specifies 3 nodes: the text 'it/It' and subject function, an adjective (JJ) and a complement clause (ccomp), where node #2 dominates the other two. This catches cataphoric cases like "It is ADJ that …" and assigns a function 'cata' which can be handled later by the system for inclusion/exclusion in coreference resolution. The rule in the second column is useful for the OntoNotes scheme, which considers ages after a comma to be coreferent appositions, i.e. in:

(9) [*Mr. Bromwich*], [*35*]

The age is seen as elliptical for something like 'a 35 year old'. The rule finds a proper noun (NNP), comma and a number in sequence, where node #1 dominates node #3, and sets the function of #3 to 'appos' for an apposition.

	JJ-that-cataphora	*age-appos*

toks	text=/^[Ii]t$/& func=/nsubj/;
	pos=/JJ/;func=/ccomp/
		pos=/^NNP$/;
		text=/^,$/;
		text=/^[1-9][0-9]*$/

rels	#2>#1;#2>#3	#1.#2.#3;#1>#3
acts	#3:func=cata	#3:func=appos

Table 2: Some dependency edit rules.

3.4 Coreference Rules

Like all other aspects of the system, coreference matching is done by way of configurable rules of the form:

C = <*ANA*, *ANT*, *DIR*, *DIST*, *PROP*>

Where *ANA* and *ANT* are feature constraints on the anaphor and the antecedent, *DIR* is the search direction (back, or forward for cataphora), *DIST* is the maximum distance in sentences to search for a match and *PROP* is the direction of feature propagation once a match is made, if any. Feature constraints include entity type/subclass, definiteness, NP-form (common/proper/pronoun), cardinality (numerical modifiers or amount of members in a coordination), and features of the head token, as well as existence/non-existence of certain modifiers or parents in a head token's dependency graph.

Rules are consulted in order, similarly to the sieve approach of Lee et al. (2013), so that the most certain rules are applied first. Every mention has only one antecedent (a mention-pair, or mention-synchronous model, like Durrett and Klein but unlike Lee et al.), so that subsequent matching can be skipped, but some aspects of a mention-cluster or 'entity-mention' model (cf. Rahman & Ng 2011) are also implemented, in that antonym modifier checks are applied to the entire chain.

The first rule in Table 3, which illustrates a very 'safe' strategy, searches for proper noun markables with identical text (=$1) in the previous 100 sentences, since these are almost always coreferent.

ANA (1)	*ANT* (2)	*DIR*	*DIST*	*PROP*

form=/proper/	form=/proper/
	text=$1
		←	100	none

lemma=/one/	form!=/proper/ mod=$1	←	4	→
text=/(his|her|its)/	form!=/pronoun/	→	0	←

Table 3: Coreference matching rules.

The middle rule looks for a mention headed by 'one' with the same modifier as its antecedent within 4 sentences, matching cases like (10). Finally the last rule attempts to match a possessive pronoun (which has not saturated its antecedent yet) to a nominal subject later on in the sentence, matching (11). This is the last rule of currently 27 in the model tested below, which were ordered based on linguistic intuition.

(10) [*the current flag*] … *the new flag* ... [*the current one*]
(11) *In* [*her*] *speech*, [*the chairwoman*] *said*…

Once two mentions match a rule, they are compared for clashing entity classes, modifiers, agreement and cardinality. Matches from a certain rule are ranked by a weighted score incorporating the dependency based entity identification certainty (e.g. how certain we are that a pronoun refers to a LOCATION), distance in sentences and in tokens, as well as a built-in bias to prefer subject and PERSON antecedents where possible. The one-pass, chain linking nature of the process means that, like Durrett & Klein's (2013) system, resolution is efficient, requires no pruning, and scales linearly with text length. The system is quite fast, taking about 2.5 seconds for an average Wall Street Journal document of about 700 tokens on an Intel Core i7 laptop.

4 Evaluation

4.1 Data

Since our system takes pure dependency parser input, gold syntax information and explicit data about speakers from spoken data are not currently integrated into our evaluation. We therefore focus on newswire material and Wiki data, for which we can also expect reasonable parsing performance. We evaluate our system on two datasets: Wall Street Journal data from OntoNotes (V5), and data from GUM (V2.1), a small corpus with texts from four Wiki based Web genres including not only news data, but also interviews, how-to guides and travel guides. Data from the WSJ corpus test section 23 will represent a proxy for an in-domain but out-of-training-data example for parser input. Good performance on both data sets would indicate that the system is able to adapt to different annotation schemes successfully.

Beyond differences in domain (WSJ reporting/Wiki genres), purpose (news and several other text types in GUM), and time (early 90's vs. 2010-2015), the schemes for the two datasets we use differ substantially, which we also expect to affect system evaluation (cf. Recasens & Hovy 2010). Table 4 gives an overview of coreference types across the corpora.

	GUM		**WSJ**	
	train	**test**	**train**	**test**
documents	46	8	540	57
tokens	37758	6321	322335	33306
nominals	11677	1933	104505	13162
coreference	7621	1294	38587	3642
- bridging	488	112	--	--
- predicative	71	14	--	--
- cataphora	52	3	--	--
- compound	506	71	--	--

Table 4: Coreference in GUM and WSJ.

GUM contains substantially more coreference annotation, despite having a very similar amount of nominal heads per token. The GUM training partition is roughly the size of the WSJ test data (section 23), at 37.7K to 33.3K tokens, and they contain similar amounts of nominal heads (11-13K). However, there are almost twice as many coreferring entities in GUM. Several differences in guidelines lead to this:

- All compound modifiers and most predicatives are candidates for coreference
- Cataphora and bridging are annotated (though we ignore bridging in the evaluation below)
- Indefinite or generic markables may have antecedents (cf. the *program trading* case in (3) above)
- Relative clauses are left outside markables, meaning backreference to the head in a clause is annotated ([*a man*] *who lost* [*his*]…)
- Recurring coordinations corefer even if they have no aggregate mention ([[*Jack*]i *and* [*Jill*]$_i$]$_k$.. [[*Jack*]$_i$ *and* [*Jill*]$_j$]$_k$; even if there is no [*they*]$_k$)
- Singletons are markables for entity type annotation in GUM, encouraging annotators who simultaneously code coreference to consider as many options as possible (although singletons are not counted in the coreference count)

Although inclusion of cataphora, bridging, predicatives and compound modifiers increases the coreference count, these are only responsible for about

1,100 cases in the training data, accounting for about 1/3 of the surplus compared to WSJ. This suggests that the greater portion of the difference is explained by indefinites, coordinate mentions and a general tendency to annotate more 'promiscuously' in GUM as compared to WSJ, as well as possible domain differences (e.g. how-to guides are rich in lists of ingredients that are mentioned repeatedly). Since a single coreferent pair contributes two coreferring entities, the effects of such binary pairs not present in OntoNotes can quickly add up.

4.2 Experimental setup

We compare our configurable rule based approach to two recent systems: Stanford's dcoref component of CoreNLP (Lee et al. 2013), version 3.6.0, and the Berkeley Coreference Resolution System (Durrett & Klein 2013), version 1.1. For both systems we used the recommended settings as of February 2016, and for the Berkeley system we used the 'joint' NER and coreference model (Durrett & Klein 2014) based on Durrett's recommendations (p.c.). In all cases, testing with other settings produced worse results on both datasets.

Since it is not reasonable to expect systems designed around schemes such as OntoNotes to perform well on GUM data, our main goal is to look at the impact of the scheme on performance for our system and less configurable ones. This is especially interesting considering the fact that there is insufficient training data to address the GUM scheme with a machine learning approach. We are also interested in how much of a difference the scheme will make, on the assumptions that high precision in particular should still carry over to settings where more annotation density is expected. None of the systems attempt to resolve bridging, so we will leave the bridging data out of the evaluation: only cases of the GUM coreference labels corresponding to anaphora, lexical coreference and apposition are included.[7]

Although our coreference resolution is rule-based, we nevertheless divide both datasets into training and test data, which means that gazetteer data, including dependency to entity type mappings, as well as 'is-a' data, may be harvested for our system from the training portions, but not from the test portions. Since we do not have gold dependency data to compare to the gold constituent parses in OntoNotes[8], we evaluate all systems on automatically parsed data using the CoreNLP pipeline for dcoref (including the Stanford Parser) and the Berkeley system's built in pipeline for the joint Entity Resolution System. Dependency parses for our system are generated using the Stanford Parser.

4.3 Results

Table 5 gives precision and recall for mention detection, while Table 6 shows coreference resolution performance according to several measures calculated using the official CoNLL scorer (version 8.01, see Pradhan et al. 2014).

	GUM			WSJ		
	R	**P**	**F1**	**R**	**P**	**F1**
xrenner	74.38	63.97	**68.78**	63.86	63.79	**63.83**
dcoref	45.77	68.01	54.72	57.30	60.26	58.74
berkeley	40.14	70.15	51.06	53.45	67.13	59.52

Table 5: Mention detection in GUM and WSJ.

Since dcoref and the Berkeley system only output coreferent mentions (in keeping with the absence of singletons in OntoNotes), mention detection performance is tightly linked to coreference resolution. On both datasets, xrenner has the highest recall, but on GUM it has the lowest precision and on WSJ the second lowest. This is likely related to the fact that under the GUM scheme, virtually all nominals (notably common noun compound modifiers) are candidates for coreference, and many are mentioned multiple times: for each re-mentioned compound, the modifier is likely to be caught as a nested coreferent markable, even if it is non-referential, unless the entire compound is flagged as 'atomic' by lexical resources. Based on 71 cases in the gold data, our precision against compound modifiers judged as referential and coreferring by GUM annotators, is 61%, and recall is

[7] More specifically, the OntoNotes 'IDENT' type subsumes GUM's 'ana' and 'coref' types, and GUM's 'appos' label mirrors OntoNotes appositions. We do not distinguish the label type in the evaluation below: only the correct coreference group IDs.

[8] An anonymous reviewer has asked whether constituent trees automatically converted using CoreNLP could be used as gold data: although we initially had the same expectation, it turns out that automatically converted data contains rather many errors, including many dependencies remaining underspecified as 'dep', and some being attached incorrectly as well.

	MUC			B^3			CEAF-e			mean
GUM	**R**	**P**	**F1**	**R**	**P**	**F1**	**R**	**P**	**F1**	**F1**
xrenner	57.12	54.83	55.95	52.01	46.48	49.09	50.27	39.87	44.47	**49.84**
dcoref	35.22	57.25	43.61	25.64	50.53	34.02	33.18	39.03	35.87	37.83
berkeley	40.67	71.77	51.92	27.76	60.65	38.09	29.14	52.17	37.40	42.47
WSJ	**R**	**P**	**F1**	**R**	**P**	**F1**	**R**	**P**	**F1**	**F1**
xrenner	49.47	50.89	50.17	41.13	46.38	43.60	46.17	42.91	44.48	**46.08**
dcoref	46.77	50.50	48.56	36.41	45.81	40.57	39.93	39.48	39.70	42.94
berkeley	45.07	54.25	49.23	37.30	46.81	41.52	35.21	49.46	41.13	43.96

Table 6: Coreference precision and recall on GUM and WSJ plain text data for three systems.

at 66%, which we consider to be a good result. Only very few compound modifiers are found other than by lexical identity, though there are some 'is-a' cases, such as the false negative in (12). Indeed, the most frequent reason for a false positive is identical modifiers not judged by annotators to be referential, as in (13).

(12) [*a* [*Mets*]$_i$ *fan*] … *cheer* [*the team*]$_i$
(13) [[*carbon*] *dioxide*] … [[*carbon*] *dioxide*]

Human annotators consider 'carbon dioxide' to be atomic, with 'carbon' not being a separate, referential entity; for the system, however, the identical, matching modifier noun is considered a good match under the GUM scheme. The other two systems have no chance of finding these, hence the lower recall and higher precision.

For cataphora and predicatives, we have much fewer cases: our system detects half of the 14 predicatives annotated in the test set, but none of the 3 cataphora in the gold standard. For the predicatives, 3 of the 7 errors are due to parser errors. For example, in the following case, the predicate 'home', annotated as coreferent with 'York', was parsed as an adverbial modifier, with the 'to'-PP parsed as the predicate:

(14) [*York*] *was* [*home first to the Ninth Legion and later the sixth*]

Such examples are likely to throw off the internal predicative recognition used by other systems as well. The remaining mistakes were caused by agreement errors (plural-singular), illustrated here:

(15) [*brains*] *is* [*the greater producer of wealth*]

For cataphora our rules were unlucky in the test set: the one case of fairly normal cataphora was passive (16), which our rules did not account for. The other cases had the form in (17), where within-clause 1st:3rd person mismatch interfered.

(16) [*it*] *being said* [*that you can see the bottom*]
(17) [*my*] *name is* [*Frank*]

Arguably in the latter case, the gold annotation is incorrect, since although the speaker is 'Frank', it's not clear 'Frank' as a name constitutes a mention of the entity. Even if accepted, this case is marginal for consideration under the heading cataphora.

For WSJ data, we excluded non-proper noun compound modifiers from the eligible markable heads, by adding the appropriate POS tags (*NN*, *NNS*) and function labels (CSD's *nn*) to our configuration, and ruled out predicatives and cataphora in the same way. As a result, precision on WSJ data is between the other two systems, while recall is still higher. The higher recall is due to some more aggressive strategies taken by our configuration, including: allowing new modifiers on later mentions (which dcoref avoids, following the tendency for no new modifiers identified in Fox 1993); a large 'is-a' table based on PPDB for non-identical lexical heads; and specific patterns, such as rules for phrases like 'the new one' based on identical modifiers, or verbal coreference based on identical stems (i.e. cases like [*required*] … [*the requirement*]).

Performance on coreference resolution for WSJ is also good, despite this being a rather difficult target (note that F-scores for both dcoref and the Berkeley system are well below the 60%+ F-scores reported for the entirety of OntoNotes, based on gold parse data, see Durrett & Klein 2013, Lee et al. 2013). Although our rules for the WSJ configuration prohibit indefinite or generic anaphors, the aggressive matching strategy sees gains over other systems mainly because of a rise in recall, with comparatively smaller hits to precision, depending on the metric (e.g. the Berkeley system has higher precision for CEAF, but xrenner always has the highest recall, and the highest F1 score in total). Some of the hits to precision are mitigated by safeguards not used by other systems, such as the categorical antonym modifier list (preventing [*the good news*] = [*the bad news*]) and cardinality matching ([*five other countries*] ≠ [*17 other countries*]). While the Berkeley system utilizes these cues indirectly via training data, number tokens are varied and sparse, but all number forms have a categorical mismatch effect on our system. By contrast, this information is not used by the dcoref sieves.

In addition, for high coverage classes, including geolocations, financial companies, newspapers and others, fine-grained entity recognition helps catch more is-a cases, such as [*the People's Daily*] … [*the newspaper*]. By appearing in Freebase as a newspaper, such entities are included under the class ORGANIZATION, subclass NEWSPAPER, thereby allowing subclass specific matching for 'newspaper'. This type of information is not captured by more coarse-grained, ORGANIZATION level NER.

5 Discussion

The results above indicate that a rule based approach backed by rich lexical data can perform well on disparate text types and annotation schemes. By relegating the large majority of system behaviors to configuration files, we are able to adjust to rather different annotation guidelines and achieve good performance on different corpora. This is facilitated by the use of dependency input, since many of the rule behaviors, including mention border definitions, can be captured in terms of dependency functions and chains. At the same time, the lack of gold dependency data to test on means that we cannot currently compare performance to gold constituent based results: this is a major goal for our planned future work, which will require careful manual correction of converted constituent data.

Some of the more challenging coreference phenomena we have attempted to model are addressable in the configurable approach: using the direction parameter for coreference rules, configurable dependency re-wiring, and a cascaded, high-precision-rule-first approach, we were able to find predicate markables and compound modifiers with high accuracy and without fatally lowering precision. This is because purely syntactic cases such as 'it is X that Y' are caught by the dependency graph analysis, high certainty cases such as reflexives and appositions are dealt with first, and other less certain cases are only applied as 'last ditch efforts', e.g. matching '*in* [*his*] *speech* [*Mr. X*] *said*' (only used if 'his' remains without an antecedent).

A major caveat for our approach is the need for domain specific lexical data. The fine-grained entity approach is not usable with leading coarse-grained NER software, meaning that high-quality lexical resources, such as the Freebase and PPDB data, are crucial. This means that while we do not require training data to change the coreference matching behavior of the system, we would need a substantial investment in new lexical data to extend to new text types and languages. We have also ordered our rules based on linguistic intuition, which may not be optimal. In future work we intend to test other permutations of our rule orders, following the approach of Lee et al. (2013: 905-906).

We are currently in the process of building models for German, based on the scheme of the largest available corpus (TüBa-D/Z, Telljohann et al. 2015), and for Coptic, an ancient low-resource language with rather limited domain vocabulary (religious texts). We hope to be able to extend our methods to these and other languages successfully by exploiting the configurable approach to change the system's behavior and adapting it to tagging and parsing input for each language as required.

Acknowledgments

We would like to thank Dan Simonson and three anonymous reviewers for valuable comments on previous versions of this paper, as well as Emma Manning for her help in creating the resources used in the evaluation. The usual disclaimers apply.

References

Emilia Apostolova, Noriko Tomuro, Pattanasak Mongkolwat and Dina Demner-Fushman. 2012. Domain Adaptation of Coreference Resolution for Radiology Reports. In *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing (BioNLP 2012)*. Montreal, 118–121.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *6th International Semantic Web Conference*. Busan, South Korea, 11–15.

BBN Technologies. 2007. *Co-reference Guidelines for English OntoNotes. Version 6.0*. Available online at: `http://www.ldc.upenn.edu/Catalog/docs/LDC2007T21/coreference/english-coref.pdf`.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. Vancouver, 1247–1250.

Danqi Chen and Christopher D. Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar, 740–750.

Greg Durrett and Dan Klein. 2013. Easy Victories and Uphill Battles in Coreference Resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*. Seattle, 1971–1982.

Greg Durrett and Dan Klein. 2014. A Joint Model for Entity Analysis: Coreference, Typing, and Linking. *Transactions of the Association for Computational Linguistics* 2:477–490.

Christiane Fellbaum (ed.). 1998. *WordNet: An Electronic Lexical Database*. MIT Press: Cambridge, MA.

Barbara A. Fox. 1993. *Discourse Structure and Anaphora: Written and Conversational English*. Cambridge: Cambridge University Press.

Juri Ganitkevitch, Benjamin Van Durme and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of NAACL-HLT*. Atlanta, GA, 758–764.

Aria Haghighi and Dan Klein. 2009. Simple Coreference Resolution with Rich Syntactic and Semantic Features. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP 2009)*. Singapore, 1152–1161.

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

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.

Marie-Catherine de Marneffe and Christopher D. Manning. 2013. *Stanford Typed Dependencies Manual*. Stanford University, Technical Report.

Massimo Poesio and Ron Artstein. 2008. Anaphoric Annotation in the ARRAU Corpus. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odjik, Stelios Piperidis and Daniel Tapias (eds.), *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC-2008)*. Marrakech, 1170–1174.

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*. Baltimore, MD, 30–35.

Altaf Rahman and Vincent Ng. 2011. Narrowing the Modeling Gap: A Cluster-Ranking Approach to Coreference Resolution. *Journal of Artificial Intelligence Research* 40(1):469–521.

Marta Recasens and Eduard Hovy. 2010. Coreference Resolution across Corpora: Languages, Coding Schemes, and Preprocessing Information. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, 1423–1432.

Utpal Kumar Sikdar, Asif Ekbal, Sriparna Saha, Olga Uryupina and Massimo Poesio. 2013. Anaphora Resolution for Bengali: An Experiment with Domain Adaptation. *Computación y Sistemas* 17(2):137–146.

Heike Telljohann, Erhard W. Hinrichs, Sandra Kübler, Heike Zinsmeister and Kathrin Beck. 2015. *Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z)*. Seminar für Sprachwissenschaft, Universität Tübingen, Technical Report.

Amir Zeldes. 2016. The GUM Corpus: Creating Multilayer Resources in the Classroom. *Language Resources and Evaluation*. Available online at: `http://dx.doi.org/10.1007/s10579-016-9343-x`.

Shanheng Zhao and Hwee Tou Ng. 2014. Domain Adaptation with Active Learning for Coreference Resolution. In *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis, EACL 2014*. Gothenburg, Sweden, 21–29.