Solving Hard Coreference Problems

Haoruo Peng* and **Daniel Khashabi*** and **Dan Roth**
University of Illinois, Urbana-Champaign
Urbana, IL, 61801
{hpeng7,khashab2,danr}@illinois.edu

Abstract

Coreference resolution is a key problem in natural language understanding that still escapes reliable solutions. One fundamental difficulty has been that of resolving instances involving pronouns since they often require deep language understanding and use of background knowledge. In this paper we propose an algorithmic solution that involves a new representation for the knowledge required to address hard coreference problems, along with a constrained optimization framework that uses this knowledge in coreference decision making. Our representation, Predicate Schemas, is instantiated with knowledge acquired in an unsupervised way, and is compiled automatically into constraints that impact the coreference decision. We present a general coreference resolution system that significantly improves state-of-the-art performance on hard, *Winograd*-style, pronoun resolution cases, while still performing at the state-of-the-art level on standard coreference resolution datasets.

1 Introduction

Coreference resolution is one of the most important tasks in *Natural Language Processing* (NLP). Although there is a plethora of works on this task (Soon et al., 2001a; Ng and Cardie, 2002a; Ng, 2004; Bengtson and Roth, 2008; Pradhan et al., 2012; Kummerfeld and Klein, 2013; Chang et al., 2013), it is still deemed an unsolved problem due to intricate and ambiguous nature of natural language text. Existing methods perform particularly poorly on pronouns, specifically when gender or plurality information cannot help. In this paper, we aim to improve coreference resolution by addressing these hard problems. Consider the following examples:

Ex.1 *[A bird]$_{e_1}$ perched on the [limb]$_{e_2}$ and [it]$_{pro}$ bent.*

Ex.2 *[Robert]$_{e_1}$ was robbed by [Kevin]$_{e_2}$, and [he]$_{pro}$ is arrested by police.*

In both examples, one cannot resolve the pronouns based on only gender or plurality information. Recently, Rahman and Ng (2012) gathered a dataset containing 1886 sentences of such challenging pronoun resolution problems (referred to later as the *Winograd* dataset, following Winograd (1972) and Levesque et al. (2011)). As an indication to the difficulty of these instances, we note that a state-of-the-art coreference resolution system (Chang et al., 2013) achieves precision of 53.26% on it. A special purpose classifier (Rahman and Ng, 2012) trained on this data set achieves 73.05%. The key contribution of this paper is a general purpose, state-of-the-art coreference approach which, at the same time, achieves precision of 76.76% on these hard cases.

Addressing these hard coreference problems requires significant amounts of background knowledge, along with an inference paradigm that can make use of it in supporting the coreference decision. Specifically, in Ex.1 one needs to know that "a limb bends" is more likely than "a bird bends". In Ex.2 one needs to know that the *subject* of the verb "rob" is more likely to be the *object* of "arrest" than the *object* of the verb "rob" is. The knowledge required is, naturally, centered around

* These authors contributed equally to this work.

the key predicates in the sentence, motivating the central notion proposed in this paper, that of *Predicate Schemas*. In this paper, we develop the notion of *Predicate Schemas*, instantiate them with automatically acquired knowledge, and show how to compile it into constraints that are used to resolve coreference within a general *Integer Linear Programming* (ILP) driven approach to coreference resolution. Specifically, we study two types of Predicate Schemas that, as we show, cover a large fraction of the challenging cases. The first specifies one predicate with its subject and object, thus providing information on the subject and object preferences of a given predicate. The second specifies two predicates with a semantically shared argument (either subject or object), thus specifies role preferences of one predicate, among roles of the other. We instantiate these schemas by acquiring statistics in an unsupervised way from multiple resources including the Gigaword corpus, Wikipedia, Web Queries and polarity information.

A lot of recent work has attempted to utilize similar types of resources to improve coreference resolution (Rahman and Ng, 2011a; Ratinov and Roth, 2012; Bansal and Klein, 2012; Rahman and Ng, 2012). The common approach has been to inject knowledge as features. However, these pieces of knowledge provide relatively strong evidence that loses impact in standard training due to sparsity. Instead, we compile our Predicate Schemas knowledge automatically, at inference time, into constraints, and make use of an ILP driven framework (Roth and Yih, 2004) to make decisions. Using constraints is also beneficial when the interaction between multiple pronouns is taken into account when making global decisions. Consider the following example:

Ex.3 *[Jack]*$_{e_1}$ *threw the bags of [John]*$_{e_2}$ *into the water since [he]*$_{pro_1}$ *mistakenly asked [him]*$_{pro_2}$ *to carry [his]*$_{pro_3}$ *bags.*

In order to correctly resolve the pronouns in Ex.3, one needs to have the knowledge that "*he* asks *him*" indicates that *he* and *him* refer to different entities (because they are subject and object of the same predicate; otherwise, *himself* should be used instead of *him*). This knowledge, which can be easily represented as constraints during inference, then impacts other pronoun decisions in a global decision with respect to all pronouns: pro_3 is likely to be different from pro_2, and is likely to refer to e_2. This type of inference can be easily represented as a constraint during inference, but hard to inject as a feature.

We then incorporate all constraints into a general coreference system (Chang et al., 2013) utilizing the mention-pair model (Ng and Cardie, 2002b; Bengtson and Roth, 2008; Stoyanov et al., 2010). A classifier learns a pairwise metric between mentions, and during inference, we follow the framework proposed in Chang et al. (2011) using ILP.

The main contributions of this paper can be summarized as follows:

1. We propose the Predicate Schemas representation and study two specific schemas that are important for coreference.
2. We show how, in a given context, Predicate Schemas can be automatically compiled into constraints and affect inference.
3. Consequently, we address hard pronoun resolution problems as a standard coreference problem and develop a system[1] which shows significant improvement for hard coreference problems while achieving the same state-of-the-art level of performance on standard coreference problems.

The rest of the paper is organized as follows. We describe our Predicate Schemas in Section 2 and explain the inference framework and automatic constraint generation in Section 3. A summary of our knowledge acquisition steps is given in Section 4. We report our experimental results and analysis in Section 5, and review related work in Section 6.

2 Predicate Schema

In this section we present multiple kinds of knowledge that are needed in order to improve hard coreference problems. Table 1 provides two example sentences for each type of knowledge. We use m to refer to a mention. A mention can either be an entity e or a pronoun pro. $pred_m$ denotes the predicate of m (similarly, $pred_{pro}$ and $pred_e$ for pronouns and entities, respectively). For instance, in sentence 1.1 in Table 1, the predicate of e_1 and e_2

[1] Available at http://cogcomp.cs.illinois.edu/page/software_view/Winocoref

Category	#	Sentence
1	1.1	*[The bird]$_{e_1}$ perched on the [limb]$_{e_2}$ and [it]$_{pro}$ bent.*
	1.2	*[The bee]$_{e_1}$ landed on [the flower]$_{e_2}$ because [it]$_{pro}$ had pollen.*
2	2.1	*[Bill]$_{e_1}$ was robbed by [John]$_{e_2}$, so the officer arrested [him]$_{pro}$.*
	2.2	*[Jimbo]$_{e_1}$ was afraid of [Bobbert]$_{e_2}$ because [he]$_{pro}$ gets scared around new people.*
3	3.1	*[Lakshman]$_{e_1}$ asked [Vivan]$_{e_2}$ to get him some ice cream because [he]$_{pro}$ was hot.*
	3.2	*Paula liked [Ness]$_{e_1}$ more than [Pokey]$_{e_2}$ because [he]$_{pro}$ was mean to her.*

Table 1: Example sentences for each schema category. The annotated entities and pronouns are hard coreference problems.

Type	Schema form	Explanation of examples from Table 1
1	$pred_m\,(m, a)$	Example 1.2: It is enough to know that: $\mathcal{S}$ (have $(m =$ [the flower], $a =$ [pollen])) $>$ $\mathcal{S}$ (have $(m =$ [the bee], $a =$ [pollen]))
2	$pred_m\,(m, a)\,\vert\widehat{pred}_m\,(m, \widehat{a})\,, cn$	Example 2.2: It is enough to know that: $\mathcal{S}$ (*be afraid of* $(m = *, a = *)\,\vert$*get scared* $(m = *, \widehat{a} = *)\,$, because) $>$ $\mathcal{S}$ (*be afraid of* $(a = *, m = *)\,\vert$*get scared* $(m = *, \widehat{a} = *)\,$, because)

Table 2: Predicate Schemas and examples of the logic behind the schema design. Here $*$ indicates that the argument is dropped, and $\mathcal{S}(.)$ denotes the scoring function defined in the text.

Type 1	$\mathcal{S}\,(pred_m\,(m, a))$ $\mathcal{S}\,(pred_m\,(a, m))$ $\mathcal{S}\,(pred_m\,(m, *))$ $\mathcal{S}\,(pred_m\,(*, m))$
Type 2	$\mathcal{S}\left(pred_m\,(m, a)\,\vert\widehat{pred}_m\,(m, \widehat{a})\,, cn\right)$ $\mathcal{S}\left(pred_m\,(a, m)\,\vert\widehat{pred}_m\,(m, \widehat{a})\,, cn\right)$ $\mathcal{S}\left(pred_m\,(m, a)\,\vert\widehat{pred}_m\,(\widehat{a}, m)\,, cn\right)$ $\mathcal{S}\left(pred_m\,(a, m)\,\vert\widehat{pred}_m\,(\widehat{a}, m)\,, cn\right)$ $\mathcal{S}\left(pred_m\,(m, *)\,\vert\widehat{pred}_m\,(m, *)\,, cn\right)$ $\vdots$

Table 3: Possible variations for scoring function statistics. Here $*$ indicates that the argument is dropped.

is $pred_{e_1} = pred_{e_2} =$"perch on". cn refers to the discourse connective (cn="and" in sentence 1.1). a denotes an argument of $pred_m$ other than m. For example, in sentence 1.1, assuming that $m = e_1$, the corresponding argument is $a = e_2$.

We represent the knowledge needed with two types of Predicate Schemas (as depicted in Table 2). To solve the assignment of [it]$_{pro}$ in sentence 1.1, as mentioned in Section 1, we need the knowledge that "a limb bends" is more reasonable than "a bird bends". Note that the predicate of the pronoun is playing a key role here. Also the entity mention itself is essential. Similarly, for sentence 1.2, to resolve [it]$_{pro}$, we need the knowledge that "bee had pollen" is more reasonable than "flower had pollen". Here, in addition to entity mention and the predicate (of the pronoun), we need the argument which shares the predicate with the pronoun. To formally define the type of knowledge needed we denote it with "$\boldsymbol{pred_m(m, a)}$" where m and a are a mention and an argument, respectively[2]. We use $\mathcal{S}(.)$ to denote the score representing how likely the combination of the predicate-mention-argument is. For each schema, we use several variations by either changing the order of the arguments (*subj.* vs *obj.*) or dropping either of them. We score the various Type 1 and Type 2 schemas (shown in Table 3) differently. The first row of Table 2 shows how Type 1 schema is being used in the case of Sentence 1.2.

For sentence 2.2, we need to have the knowledge that the *subject* of the verb phrase "be afraid of" is more likely than the *object* of the verb phrase "be afraid of" to be the *subject* of the verb phrase "get scared". The structure here is more complicated than that of Type 1 schema. To make it clearer, we analyze sentence 2.1. In this sentence, the *object* of "be robbed by" is more likely than the *subject*

[2]Note that the order of m and a relative to the predicate is a critical issue. To keep things general in the schemas definition, we do not show the ordering; however, when using scores in practice the order between a mention and an argument is a critical issue.

of the verb phrase "be robbed by" to be the *object* of "the officer arrest". We can see in both examples (and for the Type 2 schema in general), that both predicates (the entity predicate and the pronoun predicate) play a crucial role. Consequently, we design the Type 2 schema to capture the interaction between the entity predicate and the pronoun predicate. In addition to the predicates, we may need mention-argument information. Also, we stress the importance of the discourse connective between entity mention and pronoun; if in either sentence 2.1 or 2.2, we change the discourse connective to "although", the coreference resolution will completely change. Overall, we can represent the knowledge as "$\boldsymbol{pred_m}\,(\boldsymbol{m}, \boldsymbol{a})\,|\widehat{\boldsymbol{pred_m}}\,(\boldsymbol{m}, \widehat{\boldsymbol{a}})\,, \boldsymbol{cn}$". Just like for Type 1 schema, we can represent Type 2 schema with a score function for different variations of arguments (lower half of Table 3). In Table 2, we exhibit this for sentence 2.2.

Type 3 contains the set of instances which cannot be solved using schemas of Type 1 or 2. Two such examples are included in Table 1. In sentence 3.1 and 3.2, the context containing the necessary information goes beyond our triple representation and therefore this instance cannot be resolved with either of the two schema types. It is important to note that the notion of Predicate Schemas is more general than the Type 1 and Type 2 schemas introduced here. Designing more informative and structured schemas will be essential to resolving additional types of hard coreference instances.

3 Constrained ILP Inference

Integer Linear Programming (ILP) based formulations of NLP problems (Roth and Yih, 2004) have been used in a board range of NLP problems and, particularly, in coreference problems (Chang et al., 2011; Denis and Baldridge, 2007). Our formulation is inspired by Chang et al. (2013). Let $\mathcal{M}$ be the set of all mentions in a given text snippet, and $\mathcal{P}$ the set of all pronouns, such that $\mathcal{P} \subset \mathcal{M}$. We train a coreference model by learning a pairwise mention scoring function. Specifically, given a mention-pair $(u, v) \in \mathcal{M}$ (u is the antecedent of v), we learn a left-linking scoring function $f_{u,v} = \mathbf{w}^\top \phi(u, v)$, where $\phi(u, v)$ is a pairwise feature vector and $\mathbf{w}$ is the weight vector. We then follow the *Best-Link* approach (Section 2.3 from Chang et al. (2011)) for inference. The ILP problem that we solve is formally defined as follows:

$$\begin{cases} \arg\max\limits_{y} \sum\limits_{u\in\mathcal{M}, v\in\mathcal{M}} f_{u,v} y_{u,v} \\ \text{s.t. } y_{u,v} \in \{0,1\}, \quad \forall u, v \in \mathcal{M} \\ \quad \sum_{u<v, u\in\mathcal{M}} y_{u,v} \leq 1, \quad \forall v \in \mathcal{M} \\ \quad \text{Constraints from Predicate Schemas Knowledge} \\ \quad \text{Constraints between pronouns.} \end{cases}$$

Here, u, v are mentions and $y_{u,v}$ is the decision variable to indicate whether or not mention u and mention v are coreferents. As the first constraint shows, $y_{u,v}$ is a binary variable. $y_{u,v}$ equals 1 if u, v are coreferents and 0 otherwise. The second constraint indicates that we only choose at most one antecedent to be coreferent with each mention v. ($u < v$ represents that u appears beore v, thus u is an antecedent of v.) In this work, we add constraints from Predicate Schemas Knowledge and between pronouns.

The Predicate Schemas knowledge provides a vector of score values $\mathcal{S}(u, v)$ for mention pairs $\{(u, v)|(u \in \mathcal{M}, v \in \mathcal{P}\}$, which concatenates all the schemas involving u and v. Entries in the score vector are designed so that the larger the value is, the more likely u and v are to be coreferents. We have two ways to use the score values: 1) Augumenting the feature vector $\phi(u, v)$ with these scores. 2) Casting the scores as constraints for the coreference resolution ILP in one of the following forms:

$$\begin{cases} \text{if } s_i(u, v) \geq \alpha_i s_i(w, v) \Rightarrow y_{u,v} \geq y_{w,v}, \\ \text{if } s_i(u, v) \geq s_i(w, v) + \beta_i \Rightarrow y_{u,v} \geq y_{w,v}, \end{cases} \quad (1)$$

where $s_i(.)$ is the i-th dimension of the score vector $\mathcal{S}(.)$ corresponding to the i-th schema represented for a given mention pair. α_i and β_i are threshold values which we tune on a development set.[3] If an inequality holds for all relevant schemas (that is, all the dimensions of the score vector), we add an inequality between the corresponding indicator variables inside the ILP.[4] As we increase the value of a

[3]For the i_{th} dimension of the score vector, we choose either α_i or β_i as the threshold.

[4]If the constraints dictated by any two dimensions of $\mathcal{S}$ are contradictory, we ignore both of them.

threshold, the constraints in (1) become more conservative, thus it leads to fewer but more reliable constraints added into the ILP. We tune the threshold values such that their corresponding scores attain high enough accuracy, either in the multiplicative form or the additive form.[5] Note that, given a pair of mentions and context, we automatically instantiate a collection of relevant schemas, and then generate and evaluate a set of corresponding constraints. To the best of our knowledge, this is the first work to use such automatic constraint generation and tuning method for coreference resolution with ILP inference. In Section 4, we describe how we acquire the score vectors $\mathcal{S}(u, v)$ for the Predicate Schemas in an unsupervised fashion.

We now briefly explain the pre-processing step required in order to extract the score vector $\mathcal{S}(u, v)$ from a pair of mentions. Define a triple structure $t_m \triangleq \textit{pred}_m(m, a_m)$ for any $m \in \mathcal{M}$. The subscript m for *pred* and a, emphasizes that they are extracted as a function of the mention m. The extraction of triples is done by utilizing the dependency parse tree from the Easy-first dependency parser (Goldberg and Elhadad, 2010). We start with a mention m, and extract its related predicate and the other argument based on the dependency parse tree and part-of-speech information. To handle multiword predicates and arguments, we use a set of hand-designed rules. We then get the score vector $\mathcal{S}(u, v)$ by concatenating all scores of the Predicate Schemas given two triples t_u, t_v. Thus, we can expand the score representation for each type of Predicate Schemas given in Table 2: 1) For Type 1 schema, $\mathcal{S}(u, v) \equiv \mathcal{S}(pred_v(m = u, a = a_v))$ [6] 2) For Type 2 schema, $\mathcal{S}(u, v) \equiv \mathcal{S}(pred_u(m = u, a = a_u)|\widehat{pred}_v(m = v, a = a_v), cn)$.

In additional to schema-driven constraints, we also apply constraints between pairs of pronouns within a fixed distance[7]. For two pronouns that are semantically different (e.g. *he* vs. *it*), they must refer to different antecedents. For two non-possesive pronouns that are related to the same predicate (e.g. *saw* *him*), they must refer to different antecedents.[8]

4 Knowledge Acquisition

One key point that remains to be explained is how to acquire the knowledge scores $\mathcal{S}(u, v)$. In this section, we propose multiple ways to acquire these scores. In the current implementation, we make use of four resources. Each of them generates its own score vector. Therefore, the overall score vector is the concatenation of the score vector from each resource: $\mathcal{S}(u, v) = [\mathcal{S}_{giga}(u, v)\ \mathcal{S}_{wiki}(u, v)\ \mathcal{S}_{web}(u, v)\ \mathcal{S}_{pol}(u, v)]$.

4.1 Gigaword Co-occurence

We extract triples $t_m \triangleq \textit{pred}_m(m, a_m)$ (explained in Section 3) from Gigaword data (4,111,240 documents). We start by extracting noun phrases using the Illinois-Chunker (Punyakanok and Roth, 2001). For each noun phrase, we extract its head noun and then extract the associated predicate and argument to form a triple.

We gather the statistics for both schema types after applying lemmatization on the predicates and arguments. Using the extracted triples, we get a score vector from each schema type: $\mathcal{S}_{giga} = [\mathcal{S}^{(1)}_{giga}\ \mathcal{S}^{(2)}_{giga}]$.

To extract scores for Type 1 Predicate Schemas, we create occurence counts for each schema instance. After all scores are gathered, our goal is to query $\mathcal{S}^{(1)}_{giga}(u, v) \equiv \mathcal{S}(pred_v(m = u, a = a_v))$ from our knowledge base. The returned score is the $log(.)$ of the number of occurences.

For Type 2 Predicate Schemas, we gather the statistics of triple co-occurence. We count the co-occurrence of neighboring triples that share at least one linked argument. We consider two triples to be neighbors if they are within a distance of three sentences. We use two heuristic rules to decide whether a pair of arguments between two neighboring triples are coreferents or not: 1) If the head noun of two arguments can match, we consider them coreferents. 2) If one argument in the first triple is a person name and there is a compatible pronoun (based on its gender and plurality information) in the second triple, they are also labeled as coreferents. We also extract the discourse connectives between triples (*because*,

[5] The choice is made based on the performance on the development set.

[6] In $pred_v(m = u, a = a_v)$ the argument and the predicate are extracted relative to v but the mention m is set to be u.

[7] We set the distance to be 3 sentences.

[8] Three cases are considered: *he-him*, *she-her*, *they-them*

$$s_{pol}(u,v) = \begin{bmatrix} \mathbf{1}\{Po(p_u) = + \text{ AND } Po(p_v) = +\} & \text{OR} & \mathbf{1}\{Po(p_u) = - \text{ AND } Po(p_v) = -\} \\ & & \mathbf{1}\{Po(p_u) = + \text{ AND } Po(p_v) = +\} \\ & & \mathbf{1}\{Po(p_u) = - \text{ AND } Po(p_v) = -\} \end{bmatrix}$$

Table 4: Extrating the polarity score given polarity information of a mention-pair (u, v). To be brief, we use the shorthand notation $p_v \triangleq pred_v$ and $p_u \triangleq pred_u$. $\mathbf{1}\{\cdot\}$ is an indicator function. $s_{pol}(u, v)$ is a binary vector of size three.

therefore, etc.) if there are any. To avoid sparsity, we only keep the mention roles (only *subj* or *obj*; no exact strings are kept). Two triple-pairs are considered different if they have different predicates, different roles, different coreferred argument-pairs, or different discourse connectives. The co-occurrence counts extracted in this form correspond to Type 2 schemas in Table 2. During inference, we match a Type 2 schema for $\mathcal{S}^{(2)}_{giga}(u, v) \equiv \mathcal{S}(pred_u(m = u, a = a_u)|\widehat{pred}_v(m = u, a = a_v), cn)$.

Our method is related, but different from the proposal in Balasubramanian et al. (2012), who suggested to extract triples using an OpenIE system (Mausam et al., 2012). We extracted triples by starting from a mention, then extract the predicate and the other argument. An OpenIE system does not easily provide this ability. Our Gigaword counts are gathered also in a way similar to what has been proposed in Chambers and Jurafsky (2009), but we gather much larger amounts of data.

4.2 Wikipedia Disambiguated Co-occurence

One of the problems with blindly extracting triple counts is that we may miss important semantic information. To address this issue, we use the publicly avaiable Illinois Wikifier (Cheng and Roth, 2013; Ratinov et al., 2011), a system that disambiguates mentions by mapping them into correct Wikipedia pages, to process the Wikipedia data. We then extract from the Wikipedia text all entities, verbs and nouns, and gather co-occurrence statistics with these syntactic *variations*: 1) *immediately after* 2) *immediately before* 3) *before* 4) *after*. For each of these variations, we get the probability and count[9] of a pair of words (e.g. probability[10]/count for "bend" *immediately following* "limb") as separate dimensions of the score vector.

Given the co-occurrence information, we get a score vector $\mathcal{S}_{wiki}(u, v)$ corresponding to Type 1 Predicate Schemas, and hence $\mathcal{S}(u, v)_{wiki} \equiv \mathcal{S}(pred_v(m = u, a = a_v))$.

4.3 Web Search Query Count

Our third source of score vectors is web queries that we implement using Google queries. We extract a score vector $\mathcal{S}_{web}(u, v) \equiv \mathcal{S}(pred_v(m = u, a = a_v))$ (Type 1 Predicate Schemas) by querying for 1) "u a_v" 2) "u $pred_v$" 3) "u $pred_v$ a_v" 4) "a_v u"[11]. For each variation of nouns (plural and singular) and verbs (different tenses) we create a different query and average the counts over all queries. Concatenating the counts (each is a separate dimension) would give us the score vector $\mathcal{S}_{web}(u, v)$.

4.4 Polarity of Context

Another rich source of information is the polarity of context, which has been previously used for Winograd schema problems (Rahman and Ng, 2012). Here we use a slightly modified version. The polarity scores are used for Type 1 Predicate Schemas and therefore we want to get $\mathcal{S}_{pol}(u, v) \equiv \mathcal{S}(pred_v(m = u, a = a_v))$. We first extract polarity values for *Po*$(pred_u)$ and *Po*$(pred_v)$ by repeating the following procedures for each of them:

- We extract initial polarity information given the predicate (using the data provided by Wilson et al. (2005)).
- If the role of the mention is *object*, we negate its polarity.
- If there is a polarity-reversing discourse connective (such as "but") preceding the predicate, we reverse the polarity.
- If there is a negative comparative adverb (such as "less", "lower") we reverse the polarity.

[9] We use the $log(.)$ of the counts here.

[10] Conditional probability of "limb" immediately following the given verb "bend".

[11] We query this only when a_v is an adjective and $pred_v$ is a to-be verb.

	# Doc	# Train	# Test	# Mention	# Pronoun	# Predictions for Pronoun
Winograd	1886	1212	674	5658	1886	1348
WinoCoref	1886	1212	674	6404	2595	2118
ACE	375	268	107	23247	3862	13836
OntoNotes	3150	2802	348	175324	58952	37846

Table 5: Statistics of *Winograd*, *WinoCoref*, *ACE* and *OntoNotes*. We give the total number of mentions and pronouns, while the number of predictions for pronoun is specific for the test data. We added 746 mentions (709 among them are pronouns) to *WinoCoref* compared to *Winograd*.

Given the polarity values *Po*($pred_u$) and *Po*($pred_v$), we construct the score vector $\mathcal{S}_{pol}(u, v)$ following Table 4.

5 Experiments

In this section, we evaluate our system for both hard coreference problems and general coreference problems, and provide detailed anaylsis on the impact of our proposed Predicate Schemas. Since we treat resolving hard pronouns as part of the general coreference problems, we extend the *Winograd* dataset with a more complete annotation to get a new dataset. We evaluate our system on both datasets, and show significant improvemnt over the baseline system and over the results reported in Rahman and Ng (2012). Moreover, we show that, at the same time, our system achieves the state-of-art performance on standard coreference datasets.

5.1 Experimental Setup

Datasets: Since we aim to solve hard coreference problems, we choose to test our system on the *Winograd* dataset[12] (Rahman and Ng, 2012). It is a challenging pronoun resolution dataset which consists of sentence pairs based on Winograd schemas. The original annotation only specifies one pronoun and two entites in each sentence, and it is considered as a binary decision for each pronoun. As our target is to model and solve them as general coreference problems, we expand the annotation to include all pronouns and their linked entities as mentions (We call this new re-annotated dataset *WinoCoref*[13]). Ex.3 in Section 1 is from the *Winograd* dataset. It originally only specifies *he* as the pronoun in question, and we added *him* and *his* as additional target pronouns. We also use two standard coreference resolution datasets *ACE*(2004) (NIST, 2004) and *OntoNotes-5.0* (Pradhan et al., 2011) for evaluation. Statistics of the datasets are provided in Table 5.

Systems	Learning Method	Inference Method
Illinois	BLMP	BLL
IlliCons	BLMP	ILP
KnowFeat	BLMP+SF	BLL
KnowCons	BLMP	ILP+SC
KnowComb	BLMP+SF	ILP+SC

Table 6: Summary of learning and inference methods for all systems. SF stands for schema features while SC represents constraints from schema knowledge.

Baseline Systems: We use the state-of-art Illinois coreference system as our baseline system (Chang et al., 2013). It includes two different versions. One employs *Best-Left-Link* (BLL) inference method (Ng and Cardie, 2002b), and we name it *Illinois*[14]; while the other uses ILP with constraints for inference, and we name it *IlliCons*. Both systems use *Best-Link Mention-Pair* (BLMP) model for training. On *Winograd* dataset, we also treat the reported result from Rahman and Ng (2012) as a baseline.

Developed Systems: We present three variations of the Predicate Schemas based system developed here. We inject Predicate Schemas knowledge as mention-pair features and retrain the system (*KnowFeat*). We use the original coreference model and Predicate Schemas knowledge as constraints during inference (*KnowCons*). We also have a combined system (*KnowComb*), which uses the schema knowledge to add features for learning as well as constraints for inference. A summary of all systems is provided in Table 6.

[12] Available at http://www.hlt.utdallas.edu/~vince/data/emnlp12/

[13] Available at http://cogcomp.cs.illinois.edu/page/data/

[14] In implementation, we use the L^3M model proposed in Chang et al. (2013), which is slightly different. It can be seen as an extension of BLL inference method.

Dataset	Metric	Illinois	IlliCons	Rahman and Ng (2012)	KnowFeat	KnowCons	KnowComb
Winograd	Precision	51.48	53.26	73.05	71.81	74.93	**76.41**
WinoCoref	AntePre	68.37	74.32	—	88.48	88.95	**89.32**

Table 7: Performance results on *Winograd* and *WinoCoref* datasets. All our three systems are trained on *WinoCoref*, and we evaluate the predictions on both datasets. Our systems improve over the baselines by over than 20% on *Winograd* and over 15% on *WinoCoref*.

Evaluation Metrics: When evaluating on the full datasets of *ACE* and *OntoNotes*, we use the widely recoginzed metrics MUC (Vilain et al., 1995), BCUB (Bagga and Baldwin, 1998), Entity-based CEAF (CEAF$_e$) (Luo, 2005) and their average. As *Winograd* is a pronoun resolution dataset, we use precision as the evaluation metric. Although *WinoCoref* is more general, each coreferent cluster only contains 2-4 mentions and all are within the same sentence. Since traditional coreference metrics cannot serve as good metrics, we extend the precision metric and design a new one called *AntePre*. Suppose there are k pronouns in the dataset, and each pronoun has $n_1, n_2, \cdots, n_k$ antecedents, respectively. We can view predicted coreference clusters as binary decisions on each antecedent-pronoun pair (linked or not). The total number of binary decisions is $\sum_{i=1}^{k} n_i$. We then meaure how many binary decisions among them are correct; let m be the number of correct decisions, then *AnrePre* is computed as: $\frac{m}{\sum_{i=1}^{k} n_i}$.

5.2 Results for Hard Coreference Problems

Performance results on *Winograd* and *WinoCoref* datasets are shown in Table 7. The best performing system is *KnowComb*. It improves by over 20% over a state-of-art general coreference system on *Winograd* and also outperforms Rahman and Ng (2012) by a margin of 3.3%. On the *WinoCoref* dataset, it improves by 15%. These results show significant performance improvement by using Predicate Schemas knowledge on hard coreference problems. Note that the system developed in Rahman and Ng (2012) cannot be used on the *WinoCoref* dataset. The results also show that it is better to compile knowledge into constraints when the knowledge quality is high than add them as features.

5.3 Results for Standard Coreference Problems

Performance results on standard *ACE* and *OntoNotes* datasets are shown in Table 8. Our *KnowComb* system achieves the same level of performance as does the state-of-art general coreference system we base it on. As hard coreference problems are rare in standard coreference datasets, we do not have significant performance improvement. However, these results show that our additional Predicate Schemas do not harm the predictions for regular mentions.

System	MUC	BCUB	CEAFe	AVG
ACE				
IlliCons	**78.17**	81.64	**78.45**	**79.42**
KnowComb	77.51	**81.97**	77.44	78.97
OntoNotes				
IlliCons	84.10	**78.30**	**68.74**	**77.05**
KnowComb	**84.33**	78.02	67.95	76.76

Table 8: Performance results on *ACE* and *OntoNotes* datasets. Our system gets the same level of performance compared to a state-of-art general coreference system.

Category	Cat1	Cat2	Cat3
Size	317	1060	509
Portion	16.8%	56.2%	27.0%

Table 9: Distribution of instances in *Winograd* dataset of each category. Cat1/Cat2 is the subset of instances that require Type 1/Type 2 schema knowledge, respectively. All other instances are put into Cat3. Cat1 and Cat2 instances can be covered by our proposed Predicate Schemas.

5.4 Detailed Analysis

To study the coverage of our Predicate Schemas knowledge, we label the instances in *Winograd* (which also applies to *WinoCoref*) with the type of Predicate Schemas knowledge required. The distribution of the instances is shown in Table 9. Our proposed Predicate Schemas cover 73% of the instances.

We also provide an ablation study on the

Schema	AntePre(Test)	AntePre(Train)
Type 1	76.67	86.79
Type 2	79.55	88.86
Type 1 (Cat1)	90.26	93.64
Type 2 (Cat2)	83.38	92.49

Table 10: Ablation Study of Knowledge Schemas on *WinoCoref*. The first line specifies the preformance for *KnowComb* with only Type 1 schema knowledge tested on all data while the third line specifies the preformance using the same model but tested on Cat1 data. The second line specifies the preformance results for *KnowComb* system with only Type 2 schema knowledge on all data while the fourth line specifies the preformance using the same model but tested on Cat2 data.

WinoCoref dataset in Table 10. These results use the best performing *KnowComb* system. They show that both Type 1 and Type 2 schema knowledge have higher precision on Category 1 and Category 2 data instances, respectively, compared to that on full data. Type 1 and Type 2 knowledge have similiar performance on full data, but the results show that it is harder to solve instances in category 2 than those in category 1. Also, the performance drop between Cat1/Cat2 and full data indicates that there is a need to design more complicated knowledge schemas and to refine the knowledge acquisition for further performance improvement.

6 Related Work

Winograd Schema: Winograd (1972) showed that small changes in context could completely change coreference decisions. Levesque et al. (2011) proposed to assemble a set of sentences which comply with Winograd's schema. Specifically, there are pairs of sentences which are identical except for minor differences which lead to different references of the same pronoun in both sentences. These references can be easily solved by humans, but are hard, he claimed, for computer programs.

Anaphora Resolution: There has been a lot of work on anaphora resolution in the past two decades. Many of the early rule-based systems like Hobbs (1978) and Lappin and Leass (1994) gained considerable popularity. The early designs were easy to understand and the rules were designed manually. With the development of machine learning based models (Connolly et al., 1994; Soon et al., 2001b; Ng and Cardie, 2002a), attention shifted to solving standard coreference resolution problems. However, many hard coreference problems involve pronouns. As Winograd's schema shows, there is still a need for further investigation in this subarea.

World Knowledge Acquisition: Many tasks in NLP (such as Textual Entailment, Question Answering, etc.) require *World Knowledge*. Although there are many existing works on acquiring them (Schwartz and Gomez, 2009; Balasubramanian et al., 2013; Tandon et al., 2014), there is still no consensus on how to represent, gather and utilize high quality *World Knowledge*. When it comes to coreference resolution, there are a handful of works which either use web query information or apply alignment to an external knowledge base (Rahman and Ng, 2011b; Kobdani et al., 2011; Ratinov and Roth, 2012; Bansal and Klein, 2012; Zheng et al., 2013). With the introduction of Predicate Schema, our goal is to bring these different approaches together and provide a coherent view.

Acknowledgments

The authors would like to thank Kai-Wei Chang, Alice Lai, Eric Horn and Stephen Mayhew for comments that helped to improve this work. This work is partly supported by NSF grant #SMA 12-09359 and by DARPA under agreement number FA8750-13-2-0008. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

References

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

N. Balasubramanian, S. Soderland, O. Etzioni, et al. 2012. Rel-grams: a probabilistic model of relations in text. In *Proceedings of the Joint Workshop on Au-*

tomatic Knowledge Base Construction and Web-scale Knowledge Extraction, pages 101–105. Association for Computational Linguistics.

N. Balasubramanian, S. Soderland, Mausam, and O. Etzioni. 2013. Generating coherent event schemas at scale. In *EMNLP*, pages 1721–1731.

M. Bansal and D. Klein. 2012. Coreference semantics from web features. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Jeju Island, South Korea, July.

E. Bengtson and D. Roth. 2008. Understanding the value of features for coreference resolution. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 294–303, Oct.

N. Chambers and D. Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, volume 2, pages 602–610. Association for Computational Linguistics.

K. Chang, R. Samdani, A. Rozovskaya, N. Rizzolo, M. Sammons, and D. Roth. 2011. Inference protocols for coreference resolution. In *Proceedings of the Annual Conference on Computational Natural Language Learning (CoNLL)*, pages 40–44, Portland, Oregon, USA. Association for Computational Linguistics.

K. Chang, R. Samdani, and D. Roth. 2013. A constrained latent variable model for coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 601–612. Association for Computational Linguistics.

X. Cheng and D. Roth. 2013. Relational inference for wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796. Association for Computational Linguistics.

D. Connolly, J. D. Burger, and D. S. Day. 1994. A machine learning approach to anaphoric reference. In *Proceedings of the International Conference on New Methods in Language Processing (NeMLaP)*. ACL.

P. Denis and J. Baldridge. 2007. Joint determination of anaphoricity and coreference resolution using integer programming. In *Proceedings of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*.

Y. Goldberg and M. Elhadad. 2010. An efficient algorithm for easy-first non-directional dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 742–750. Association for Computational Linguistics.

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

H. Kobdani, H. Schuetze, M. Schiehlen, and H. Kamp. 2011. Bootstrapping coreference resolution using word associations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 783–792. Association for Computational Linguistics.

K. J. Kummerfeld and D. Klein. 2013. Error-driven analysis of challenges in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 265–277. Association for Computational Linguistics.

S. Lappin and H. J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational linguistics*, 20(4):535–561.

H. J. Levesque, E. Davis, and L. Morgenstern. 2011. The winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*.

X. Luo. 2005. On coreference resolution performance metrics. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

Mausam, M. Schmitz, R. Bart, S. Soderland, and O. Etzioni. 2012. Open language learning for information extraction. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

V. Ng and C. Cardie. 2002a. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.

V. Ng and C. Cardie. 2002b. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

V. Ng. 2004. Learning noun phrase anaphoricity to improve conference resolution: Issues in representation and optimization. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*.

NIST. 2004. The ACE evaluation plan.

S. Pradhan, L. Ramshaw, M. Marcus, M. Palmer, R. Weischedel, and N. Xue. 2011. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Annual Conference on Computational Natural Language Learning (CoNLL)*.

S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang. 2012. CoNLL-2012 shared task: Modeling

multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Annual Conference on Computational Natural Language Learning (CoNLL)*.

V. Punyakanok and D. Roth. 2001. The use of classifiers in sequential inference. In *The Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 995–1001. MIT Press.

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

A. Rahman and V. Ng. 2011b. Coreference resolution with world knowledge. In *ACL*, pages 814–824.

A. Rahman and V. Ng. 2012. Resolving complex cases of definite pronouns: the winograd schema challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 777–789. Association for Computational Linguistics.

L. Ratinov and D. Roth. 2012. Learning-based multi-sieve co-reference resolution with knowledge. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1375–1384, Portland, Oregon, USA, June. Association for Computational Linguistics.

D. Roth and W. Yih. 2004. A linear programming formulation for global inference in natural language tasks. In Hwee Tou Ng and Ellen Riloff, editors, *Proceedings of the Annual Conference on Computational Natural Language Learning (CoNLL)*, pages 1–8. Association for Computational Linguistics.

H. A. Schwartz and F. Gomez. 2009. Acquiring applicable common sense knowledge from the web. In *Proceedings of the Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*, UMSLLS '09, pages 1–9, Stroudsburg, PA, USA. Association for Computational Linguistics.

W. M. Soon, D. C. Y. Lim, and H. T. Ng. 2001a. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics, Volume 27, Number 4, December 2001*.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001b. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544.

V. Stoyanov, C. Cardie, N. Gilbert, E. Riloff, D. Buttler, and D. Hysom. 2010. Coreference resolution with reconcile. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 156–161. Association for Computational Linguistics.

N. Tandon, G. de Melo, and G. Weikum. 2014. Acquiring comparative commonsense knowledge from the web. In *Proceedings of AAAI Conference on Artificial Intelligence*. AAAI.

M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on Message understanding*.

T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan. 2005. Opinionfinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP on interactive demonstrations*, pages 34–35. Association for Computational Linguistics.

T. Winograd. 1972. Understanding natural language. *Cognitive psychology*, 3(1):1–191.

J. Zheng, L. Vilnis, S. Singh, J. D. Choi, and A. McCallum. 2013. Dynamic knowledge-base alignment for coreference resolution. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning (CoNLL)*, pages 153–162, Sofia, Bulgaria, August. Association for Computational Linguistics.