Entity Coreference Resolution

Vincent Ng

***Abstract*—Entity coreference resolution is generally considered one of the most difficult tasks in natural language understanding. Though extensively investigated for more than 50 years, the task is far from being solved. Its difficulty stems from its reliance on sophisticated knowledge sources and inference mechanisms. Nevertheless, significant progress has been made on learning-based coreference research since its inception two decades ago. This paper provides an overview of the major milestones made in learning-based coreference research.**

***Index Terms*—text mining, natural language processing, information extraction, coreference resolution, anaphora resolution**

I. Introduction

ENTITY coreference resolution is generally considered one of the most difficult tasks in natural language processing (NLP). The task involves determining which entity mentions in a text or dialogue refer to the same real-world entity. Despite being investigated for 50 years in the NLP community, the task is still far from being solved. To better understand its difficulty, consider the following sentence:

> The Queen Mother asked Queen Elizabeth II to transform her sister, Princess Margaret, into a viable princess by summoning a renowned speech therapist, Nancy Logue, to treat her speech impediment.

A coreference system should partition the entity mentions in this sentence into three coreference chains — QE (*Queen Elizabeth II* and the first occurrence of *her*), PM (*sister*, *Princess Margaret* and the second occurrence of *her*), and NL (*a renowned speech therapist* and *Nancy Logue*) — and three singletons, *The Queen Mother*, *a viable princess*, and *speech impediment*.

While human audiences have few problems with identifying these co-referring mentions, the same is not true for automatic coreference resolvers. For instance, resolving the two occurrences of *her* in this example is challenging for a coreference resolver. To resolve the first occurrence of *her*, a resolver would determine whether it is coreferent with *The Queen Mother* or *Queen Elizabeth II*, but the portion of the sentence preceding the pronoun does not contain sufficient information for correctly resolving it. The only way to correctly resolve the pronoun is to employ the background knowledge that *Princess Margaret* is *Queen Elizabeth II*'s sister. To resolve the second occurrence of *her*, if a resolver employs the commonly-used heuristic that selects the closest grammatically compatible mention in the subject position as its antecedent, it will wrongly posit *Nancy Logue* as its antecedent. Even if the sentence did not mention that *Nancy Logue* was a speech therapist, a human would have no problem with correctly resolving the pronoun (to *Princess Margaret*), because he could easily rule out *Nancy Logue* as the correct antecedent by employing the commonsense knowledge that it does not make sense for Person A to summon Person B to treat Person B's problem.

From this example, we can see that background knowledge, which is typically difficult for a machine to acquire, plays an important role in coreference resolution. In general, however, the difficulty of coreference resolution, particularly the resolution of pronouns and common noun phrases, stems from its reliance on sophisticated knowledge sources and inference mechanisms [1]. Despite its difficulty, coreference resolution is a core task in information extraction: it is the fundamental technology for consolidating the textual information about an entity, which is crucial for essentially all NLP applications, such as question answering, information extraction, text summarization, and machine translation. For instance, given the question *When was Mozart born?*, a question-answering system should search for the answer in a set of documents retrieved by a search engine that contain the keywords in the question. If the answer appears in the sentence *He was born in Salzburg, Austria, in 27 January 1756*, the system can be sure that *27 January 1756* is the correct answer only if the pronoun *He* is coreferent with *Mozart*.

As coreference resolution is inherently a clustering task, it has received a lot of attention in the machine learning and data mining communities, where the task has been tackled under different names, such as *record linkage/matching* and *duplicate detection*. Some researchers have focused on *name matching*, where the goal is to determine whether the names appearing in two records in a database refer to the same entity. The focus on name matching effectively ignores pronoun resolution and common noun phrase resolution, which are arguably the most difficult subtasks of entity coreference resolution [2].

There is a recent surge of interest in pronoun resolution in the knowledge representation community owing to the Winograd Schema Challenge (WSC). The WSC was motivated by the following pair of sentences, which was originally used by Winograd [3] to illustrate the difficulty of NLP:

(1) The city council refused the women a permit because *they* feared violence.

(2) The city council refused the women a permit because *they* advocated violence.

Using world knowledge, humans can easily resolve the occurrences of *they* in sentences (1) and (2) to *The city council* and *the women* respectively. However, these pronouns are difficult to resolve automatically. One reason for this is that these pronouns are compatible with both candidate antecedents in number, gender, and semantic class. Another reason is that correct resolution may not be possible without understanding the two events mentioned in a sentence, but such understanding

Human Language Technology Research Institute, University of Texas at Dallas, Richardson, TX, USA; e-mail vince@hlt.utdallas.edu.

typically requires background knowledge. Levesque [4] argued that the resolution of difficult-to-resolve pronouns in *twin* sentences like these constitutes a task that can serve as an appealing alternative to the Turing Test. The WSC is currently being promoted by Commonsense Reasoning[1], so we expect to see continued progress on this task.

Our goal in this paper is to provide the reader with an overview of the major milestones made in learning-based entity coreference research since its inception 20 years ago. For a detailed treatment of this topic, we refer the reader to a recent book edited by Poesio et al. [5]. Given Levesque's aforementioned proposal that the resolution of difficult-to-resolve pronouns can serve as an appealing alternative to the Turing Test, we believe that the entity coreference task will be of interest to the general intelligence systems community.

II. Brief History

Learning-based entity coreference research was to a large extent stimulated by the public availability of coreference-annotated corpora that were produced as a result of three large-scale evaluations of coreference systems:

The MUC evaluations. The coreference evaluations conducted as part of the DARPA-sponsored MUC-6 [6] and MUC-7 [7] conferences provided the first two publicly available coreference corpora, the MUC-6 corpus (30 training texts and 30 test texts) and the MUC-7 corpus (30 training texts and 20 test texts). They also defined the coreference task that the NLP community sees today. In particular, the MUC organizers decided that the task should focus exclusively on *identity* coreference resolution, ignoring other kinds of coreference relations that would be challenging even for humans to identify, such as bridging (e.g., set-subset relations, part-whole relations). A significant byproduct of the MUC coreference evaluation was the first evaluation metric for coreference resolution, the MUC scoring metric [8]. Virtually all learning-based resolvers developed between 1995 and 2004 were trained and evaluated on the MUC corpora using the MUC metric.

The ACE evaluations. As part of NIST-sponsored ACE evaluations, which began in the late 1990s, four coreference corpora were released, namely ACE-2, ACE03, ACE04, and ACE05. To encourage multilingual coreference research, ACE04 and ACE05 were composed of coreference-annotated texts not only for English, but also for Chinese and Arabic. These two corpora were also heavily used for training and evaluation in part because they were much larger than the MUC corpora. For instance, the ACE04 and ACE05 English coreference training corpora were composed of 443 and 599 documents, respectively. Unlike MUC, which requires the identification of coreferent entities regardless of their semantic types, ACE focused on a restricted, simpler version of the coreference task, requiring that coreference chains be identified only for entities belonging to one of the ACE entity types (e.g., PERSON, ORGANIZATION, GPE, FACILITY, LOCATION). Virtually all resolvers developed between 2004 and 2010 were trained and evaluated on one of these ACE corpora.

To evaluate coreference systems in the official ACE evaluations, the ACE metric was developed, but it was never popularly used by coreference researchers. Two important scoring measures were developed during this period, namely B^3 [9] and CEAF [10].

Direct comparisons among the different coreference systems developed at that time were difficult for at least two reasons. First, different resolvers were evaluated on different corpora (ACE04 vs. ACE05) using different evaluation metrics (B^3 vs. CEAF). Second, and more importantly, they were trained and evaluated on different train-test splits of the ACE corpora, owing to the fact that the ACE organizers released only the training portion but not the official test portion of the ACE corpora. Worse still, some resolvers were evaluated on *gold* rather than *system* (i.e., automatically extracted) entity mentions [11], reporting substantially better results than end-to-end resolvers. This should not be surprising: coreference on gold mentions is a substantially simplified version of the coreference task because system mentions typically significantly outnumber gold mentions. Some of these complications were referred to as "conundrums" in entity coreference resolution and discussed in detail by Stoyanov et al. [12].

The CoNLL 2011 and 2012 shared tasks. The CoNLL 2011 [13] and 2012 [14] shared tasks focused on English and multilingual (English, Chinese, and Arabic) coreference resolution, respectively, using the OntoNotes 5.0 corpus [15] for training and evaluation. These shared tasks were important for two reasons. First, they directed researchers' attention back to the challenging *unrestricted* coreference tasks that were originally defined in MUC while providing substantially more data for training and evaluation. Second, and more importantly, they facilitated performance comparisons of different resolvers, making it possible to determine the state of the art. Specifically, they standardized not only the train-test partition of the OntoNotes corpus, but also the evaluation metric, the CoNLL metric [13], which is the unweighted average of MUC, B^3, and CEAF. Virtually all resolvers developed since 2011 were evaluated on this corpus.

III. Evaluation Measures

Designing evaluation measures for coreference resolution is by no means a trivial task. In this section, we describe the four most commonly-used coreference evaluation measures, each of which reports performance in terms of recall, precision, and F-score. Below we use the terms *coreference chains* and *coreference clusters* interchangeably. For a coreference chain C, we define $|C|$ as the number of mentions in C. *Key chains* and *system chains* refer to gold coreference chains and system-generated coreference chains, respectively. In addition, $\mathcal{K}(d)$ and $\mathcal{S}(d)$ refer to the set of gold chains and the set of system-generated chains in document d, respectively. Specifically,

$$\mathcal{K}(d) = \{K_i : i = 1, 2, \cdots, |\mathcal{K}(d)|\},$$

$$\mathcal{S}(d) = \{S_j : j = 1, 2, \cdots, |\mathcal{S}(d)|\},$$

where K_i is a chain in $\mathcal{K}(d)$ and S_j is a chain in $\mathcal{S}(d)$. $|\mathcal{K}(d)|$ and $|\mathcal{S}(d)|$ are the number of chains in $\mathcal{K}(d)$ and $\mathcal{S}(d)$, respectively.

[1] http://commonsensereasoning.org/winograd.html

A. MUC

MUC [8] is a link-based metric. Given a document d, recall is computed as the number of common links between the key chains and the system chains in d divided by the number of links in the key chains. Precision is computed as the number of common links divided by the number of links in the system chains. Below we show how to compute (1) the number of common links, (2) the number of key links, and (3) the number of system links.

To compute the number of common links, a partition $P(S_i)$ is created for each system chain S_i using the key chains. Specifically,

$$P(S_j) = \{C_j^i : i = 1, 2, \cdots, |\mathcal{K}(d)|\} \tag{1}$$

Each subset C_j^i in $P(S_i)$ is formed by intersecting S_j with K_i. Note that $|C_j^i| = 0$ if S_j and K_i have no mentions in common. Since there are $|\mathcal{K}(d)| * |\mathcal{S}(d)|$ subsets in total, the number of common links is

$$c(\mathcal{K}(d), \mathcal{S}(d)) = \sum_{j=1}^{|\mathcal{S}(d)|} \sum_{i=1}^{|\mathcal{K}(d)|} w_c(C_j^i), \tag{2}$$
$$\text{where } w_c(C_j^i) = \begin{cases} 0 & \text{if } |C_j^i| = 0; \\ |C_j^i| - 1 & \text{if } |C_j^i| > 0. \end{cases}$$

Intuitively, $w_c(C_j^i)$ can be interpreted as the "weight" of C_j^i. In MUC, the weight of a cluster is defined as the *minimum* number of *links* needed to create the cluster, so $w_c(C_j^i) = |C_j^i| - 1$ if $|C_j^i| > 0$.

The number of links in the key chains, $\mathcal{K}(d)$, is calculated as:

$$k(\mathcal{K}(d)) = \sum_{i=1}^{|\mathcal{K}(d)|} w_k(K_i), \tag{3}$$

where $w_k(K_i) = |K_i| - 1$. The number of links in the system chains, $s(\mathcal{S}(d))$, is calculated as:

$$s(\mathcal{S}(d)) = \sum_{j=1}^{|\mathcal{S}(d)|} w_s(S_j), \tag{4}$$

where $w_s(S_j) = |S_j| - 1$.

B. B^3

MUC's often-criticized weakness is that it fails to reward successful identification of singleton clusters. To address this weakness, B^3 [9] first computes the recall and precision for each mention, and then averages these per-mention values to obtain the overall recall and precision.

Let m_n be the nth mention in document d. Its recall, $R(m_n)$, and precision, $P(m_n)$, are computed as follows. Let K_i and S_j be the key chain and the system chain that contain m_n, respectively, and let C_j^i be the set of mentions appearing in both S_j and K_i.

$$R(m_n) = \frac{w_c(C_j^i)}{w_k(K_i)}, P(m_n) = \frac{w_c(C_j^i)}{w_s(S_j)}, \tag{5}$$

where $w_c(C_j^i) = |C_j^i|$, $w_k(K_i) = |K_i|$, and $w_s(S_j) = |S_j|$.

C. CEAF

While B^3 addresses the shortcoming of MUC, Luo [10] presents counter-intuitive results produced by B^3, which it attributes to the fact that B^3 may use a key/system chain more than once when computing recall and precision. To ensure that each key/system chain will be used at most once in the scoring process, his CEAF scoring metric scores a coreference partition by finding an optimal *one-to-one mapping* (or *alignment*) between the chains in $\mathcal{K}(d)$ and those in $\mathcal{S}(d)$.

Since the mapping is one-to-one, not all key chains and system chains will be involved in it. Let $\mathcal{K}_{min}(d)$ and $\mathcal{S}_{min}(d)$ be the set of key chains and the set of system chains involved in the alignment, respectively. The alignment can be represented as a one-to-one mapping function g, where

$$g(K_i) = S_j, K_i \in \mathcal{K}_{min}(d) \text{ and } S_j \in \mathcal{S}_{min}(d).$$

The score of g, $\Phi(g)$, is defined as

$$\Phi(g) = \sum_{K_i \in \mathcal{K}_{min}(D)} \phi(K_i, g(K_i)),$$

where ϕ is a function that computes the *similarity* between a gold chain and a system chain. The optimal alignment, g^*, is the alignment whose Φ value is the largest among all possible alignments, and can be computed efficiently using the Kuhn-Munkres algorithm [16].

Given g^*, the recall (R) and precision (P) of a system partition can be computed as follows:

$$R = \frac{\Phi(g^*)}{\sum_{i=1}^{|\mathcal{K}(d))|} \phi(K_i, K_i)}, P = \frac{\Phi(g^*)}{\sum_{j=1}^{|\mathcal{S}(d))|} \phi(S_j, S_j)}.$$

As we can see, at the core of CEAF is the similarity function ϕ. Luo defines two different ϕ functions, ϕ_3 and ϕ_4:

$$\phi_3(K_i, S_j) = |K_i \cap S_j| = w_c(C_j^i) \tag{6}$$

$$\phi_4(K_i, S_j) = \frac{2|K_i \cap S_j|}{|K_i| + |S_j|} = \frac{2 * w_c(C_j^i)}{w_k(K_i) + w_s(S_j)} \tag{7}$$

ϕ_3 and ϕ_4 result in mention-based CEAF (a.k.a. CEAF_m) and entity-based CEAF (a.k.a. CEAF_e), respectively.

D. BLANC

BLANC [17], a Rand-index-based coreference evaluation measure, is designed to address a major weakness shared by B^3 and CEAF: the B^3 and CEAF F-scores typically squeeze up too high when many singleton mentions are present in a document. To address this weakness, BLANC first computes recall, precision, and F-score separately for coreferent mention pairs and non-coreferent mention pairs. The BLANC recall/precision/F-score is then computed as the unweighted average of the recall/precision/F-score of the coreferent mention pairs and the recall/precision/F-score of the non-coreferent mention pairs.

IV. Models

In this section, we examine the major learning-based models for entity coreference resolution.

A. Mention-Pair Models

Despite their conceptual simplicity, mention-pair models are arguably the most influential coreference model. A mention-pair model is a binary classifier that determines whether a pair of mentions is co-referring or not. Hence, to train a mention-pair model, each training instance corresponds to a pair of mentions and is represented by *local* features encoding each of the two mentions and their relationships. Any learning algorithm can be used to train a mention-pair model, which can then be applied to classify the test instances. However, these pairwise classification decisions could violate transitivity, which is an inherent property of the coreference relation. As a result, a separate clustering mechanism, such as *single-link clustering* [18] and *best-first* clustering [2], is needed to coordinate the pairwise decisions and construct a partition. Specifically, these clustering algorithms process the mentions in a test text in a left-to-right manner. For each mention encountered, they select as its antecedent either the closest or the most probable preceding coreferent mention. No antecedent will be selected for the mention if it does not have any preceding coreferent mention.

It was around this time that Ng and Cardie [19] raised the question of whether *anaphoricity* should be modeled explicitly in coreference resolution. Anaphoricity determination is the task of determining whether a mention is *anaphoric* (i.e., it is coreferent with a preceding mention) or *non-anaphoric* (i.e., it starts a new coreference chain).

To motivate anaphoricity determination, consider the two aforementioned clustering algorithms, which do not perform anaphoricity determination explicitly. Specifically, s mention is implicitly posited as non-anaphoric if none of its preceding mentions is classified as coreferent with it. Ng and Cardie [19] hypothesize that performing anaphoricity determination prior to coreference resolution could improve the precision of a mention-pair model, as the model will only need to resolve mentions that are determined to be anaphoric by the anaphoricity model. While anaphoricity determination is by no means an easier task than coreference resolution, many years of research on the explicit modeling of anaphoricity have resulted in models that can benefit coreference. One such successful attempt was made by Denis and Baldridge [20], who perform joint inference over the outputs of two independently-trained models, the anaphoricity model and the mention-pair model.

B. Mention-Ranking Models

A major weakness of mention-pair models is that they consider each candidate antecedent of an anaphoric mention to be resolved independently of other candidate antecedents. As a result, they can only determine how good a candidate antecedent is relative to the anaphoric mention, but not how good it is relative to other candidate antecedents.

Ranking models address this weakness by allowing all candidate antecedents of a mention to be ranked *simultaneously* [21]–[23]. Since a mention ranker simply ranks candidate antecedents, it cannot determine if a mention is anaphoric. One way to address this problem is to apply an independently trained anaphoricity classifier to identify non-anaphoric mentions prior to ranking [23]. Another, arguably better, way is to jointly learn coreference and anaphoricity by augmenting the candidate set of each mention to be resolved with a dummy candidate antecedent so that the mention will be classified as non-anaphoric if it is resolved to the dummy [24].

C. Entity-Based Models

Another major weakness of mention-pair models concerns their limited *expressiveness*: they can only employ features defined on no more than two mentions. However, the information extracted from the two mentions alone may not be sufficient for making an informed coreference decision, especially if the candidate antecedent is a pronoun (which is semantically empty) or a mention that lacks descriptive information such as gender (e.g., *Clinton*).

Entity-based models aim to address the expressiveness problem. To motivate these models, consider a document that consists of three mentions: *Mr. Clinton*, *Clinton*, and *she*. A mention-pair model may determine that *Mr. Clinton* and *Clinton* are coreferent using string-matching features, and that *Clinton* and *she* are coreferent based on proximity and lack of evidence for gender and number disagreement. However, these two pairwise decisions together with transitivity imply that *Mr. Clinton* and *she* will end up in the same cluster, which is incorrect due to gender mismatch. This kind of error arises in part because the later coreference decisions are not dependent on the earlier ones. In particular, had the model taken into consideration that *Mr. Clinton* and *Clinton* were in the same cluster, it probably would not have posited that *she* and *Clinton* are coreferent. Specifically, the increased expressiveness of entity-based models stems from their ability to exploit *cluster-level* (a.k.a. *non-local*) features, which are features defined on an arbitrary subset of the mentions in a coreference cluster. In our example, it would be useful to have a cluster-level feature that encodes whether the gender of a mention is compatible with the gender of *each* of the mentions in a preceding cluster, for instance.

Many machine-learned entity-based models have been developed over the years. The most notable ones include the entity-based versions of mention-pair models and mention-ranking models. *Entity-mention* models, the entity-based version of mention-pair models, determine whether a mention is coreferent with a preceding, possibly partially-formed, *cluster* [25], [26]. Despite their improved expressiveness, early entity-mention models have not yielded particularly encouraging results. *Cluster-ranking* models, on the other hand, are the entity-based version of mention-ranking models [24]. They rank preceding clusters rather than candidate antecedents, and have been shown to outperform entity-mention models, mention-pair models, and mention-ranking models.

While the entity-based models discussed so far have all attempted to process the mentions in a test text in a left-to-right manner, *easy-first* models aim to make easy linking decisions first, and then use the information extracted from the clusters established thus far to help identify the difficult links. More specifically, an easy-first resolver is composed of a pipeline of *sieves*, each of which is composed of a set of hand-crafted or learned rules for classifying a *subset* of the mention pairs in the test set. Being an easy-first approach, the sieves in the pipeline are arranged in decreasing order of precision. Given the pipeline setup, the later sieves can exploit the decisions made by the earlier sieves. The most well-known resolver that employs an easy-first approach is arguable Stanford's resolver [27], which won the CoNLL-2011 shared task. Ratinov and Roth's easy-first resolver [28] improves Stanford's resolver by allowing earlier decisions to be overridden and corrected by later sieves.

Entity-based models are also trained by Culotta et al. [29] and Stoyanov and Eisner [30]. Specifically, they propose a "learning to cluster" approach to train coreference models to perform *agglomerative* clustering of the entity mentions, each of which is initially is in its own cluster.

D. Structured Models

Recent years have seen a popular line of work that views coreference resolution as a *structured prediction* task: rather than resolving a mention to a preceding mention/cluster, a structured model predicts a structure from which a coreference partition can be directly recovered.

The first such attempts are made by McCallum and Wellner [11] and Finley and Joachims [31], who train models to directly induce coreference partitions. Specifically, McCallum and Wellner train a log-linear model to induce a distribution over the possible partitions of a set of mentions so that the correct partition is the most probable. Finley and Joachims, on the other hand, learn to rank candidate coreference partitions by training a max-margin ranking model.

While learning to partition is a novel idea, partition-based models are not particularly popular. One reason is that they force us to classify each pair of mentions, which is not desirable as not all coreference links are equally easy to identify. Fortunately, to establish a cluster of n mentions, only $n - 1$ coreference links are needed. So, rather than learning a partition, Fernandes et al. [32] (FDM) propose learning a coreference *tree* using the links that are easy to identify, and then recovering a partition from the tree. To learn to predict coreference trees, FDM employ the latent structured voted perceptron algorithm. The model parameters are weights defined on features that are commonly-used in mention-pair models. In each iteration, the highest-scoring (i.e., maximum spanning) tree is decoded using the Chu-Liu-Edmonds algorithm [33], [34]. Their resolver achieved the highest average score over all languages in the CoNLL-2012 shared task. As noted by FDM, feature induction plays an important role in their resolver. Their feature induction method learns feature conjunctions, which are derived from the paths of a decision tree-based mention-pair model.

Seeing no reason to predict structures as complicated as trees, Durrett and Klein [35] (D&K) simplify the coreference task by proposing a model that predicts for each test document the most probable *antecedent structure*, which is a vector of antecedents storing the antecedent chosen for each mention (null if the mention is non-anaphoric) in the document. Effectively, it is a mention-ranking model, but it is trained to maximize the conditional likelihood of the correct antecedent structure given a document. Inference is easy: the most probable candidate antecedent of a mention is selected to be its antecedent independently of other mentions. One of the innovations of D&K's model is the use of a task-specific loss function. Specifically, D&K employ a loss function that is a weighted sum of the counts of three error types: the number of false anaphors, the number of false non-anaphors, and the number of wrong links. Following FDM, D&K employ feature conjunctions. Perhaps most interestingly, D&K achieved state-of-the-art performance by training their model only on conjunctions of *lexical* features.

Motivated in part by the recent successes of neural models for NLP tasks, Wiseman et al. [36] train a *neural-based* mention-ranking model which, like D&K's model, employs a task-specific loss function. However, rather than following the recent trend on training *linear* models using feature conjunctions [32], [35], [37], some of which are rather complex, Wiseman et al. pioneered using a neural network to learn *non-linear* representations of *raw* features (i.e., the original features, without any conjunctions), achieving state-of-the-art results. Most recently, Wiseman et al. [38] and Clark and Manning [39] further improved the performance of neural coreference models by incorporating cluster-based features. These are the first attempts to learn non-linear models of coreference resolution. Given their promising results, they deserve further investigations.

V. Knowledge Sources

Early learning-based coreference resolvers have relied primarily on morpho-syntactic knowledge. However, the development of large lexical knowledge bases since the late 1990s and the significant advancements made in corpus-based lexical semantics research in the past 15 years have enabled researchers to design semantic features for coreference resolution. In this section, we examine these two types of knowledge sources.

A. Morpho-syntactic Features

Morpho-syntactic features typically refer to several types of features. **String-matching features** encode whether there is an exact or partial match (e.g., head match, exact match after removing determiners) between the strings of the t-wo mentions under consideration. These features are useful because many coreferent mentions have overlaps in their strings (e.g., *Bill Clinton* and *Clinton*). **Lexical features** are created by concatenating the strings/heads of the two mentions. These features enable a learning algorithm to learn which string/head combinations are indicative of coreference relations. **Grammatical features** encode whether the two mentions are compatible with respect to various grammatical attributes such as gender and number. These features are useful because grammatical incompatibility is a strong indicator of non-coreference. Finally, **syntactic features** encode whether two mentions can be coreferent based on information extracted from syntactic parse trees. For instance, two mentions cannot be coreferent if they violate the Binding Constraints.

B. Semantic Features

Selectional preference is one of the earliest kinds of semantic knowledge exploited for coreference resolution [40]–[42]. Given a pronoun to be resolved, its governing verb, and its grammatical role, a candidate antecedent that can play the same role and be governed by the same verb is preferred. These preferences can be learned from a large corpus or from the Web, and have been used as features to improve knowledge-poor resolvers with varying degrees of success.

Another commonly-used semantic feature for coreference resolution encodes whether the two mentions involved have the same **semantic class**, where the semantic class of a common noun is determined using either WordNet [18], [43] or clusters induced from the Google n-gram corpus [44].

Knowing that *Barack Obama* is a *U. S. president* would be helpful for establishing the coreference relation between two mentions *Obama* and *the president* in a document. To this end, researchers have attempted to extract the **knowledge attributes** of a proper name from lexical knowledge bases. For instance, given a proper name, Ratinov and Roth [28] extract from Wikipedia its Wiki category, gender, and nationality, and Hajishirzi et al. [45] extract from Freebase a set of coarse-grained attributes (e.g., *person*, *location*) and more than 500 fine-grained attributes (e.g., *plant*, *attraction*, *nominee*). The major challenge in extracting attributes from these knowledge bases is entity disambiguation [46]: a proper name could be matched more than one Wikipedia page or more than one entry in YAGO and Freebase. To address this problem, Ratinov and Roth [28] employ a context-sensitive entity disambiguation system, while Hajishirzi et al. [45] propose to jointly perform coreference resolution and entity linking. Knowledge attributes can also be extracted in an unsupervised manner using hand-crafted lexico-syntactic patterns [47]. For instance, we can search for the pattern *X is a Y* in a large, unannotated corpus. The mention pairs (X,Y) that satisfy this pattern can tell us that mention X has knowledge attribute Y.

Besides the IS-A relation, other **semantic relations**, including those between common nouns, have also been used for coreference resolution. For instance, Bengtson and Roth [48] have employed as features the generic semantic relations (e.g., synonymy, hypernymy, antonymy) extracted from WordNet for two common nouns. Hearst [47] has proposed other lexico-syntactic patterns that capture different lexical semantic relations between nouns. Yang and Su [49] employ patterns *learned* from a coreference corpus that are indicative of a coreference relation.

Some words may not have a semantic relation but can still be coreferent owing to their **semantic similarity**. This observation has led Ponzetto and Strube [43] to encode features based on various measures of WordNet similarity, which have been shown to improve their baseline system.

PropBank-style *semantic roles* have also been used for coreference resolution [43]. Their use is motivated by the **semantic parallelism** heuristic: given an anaphor with semantic role r, its antecedent is likely to have role r.

While using semantic roles improves Ponzetto and Strube's resolver [43], semantic parallelism is a fairly weak indicator of coreference. For instance, if two verbs denote events that are unrelated to each other, it is not clear why their arguments should be coreferent even if they have the same semantic role. Motivated by this observation, Rahman and Ng [46] attempt to capture the notion of **event relatedness** based on whether the two predicates appear in the same *FrameNet semantic frame*, designing features that encode not only whether the two mentions have the same role but also whether their governing verbs are in the same frame.

Generally speaking, the results of employing semantic and world knowledge to improve knowledge-poor coreference resolvers are mixed. The mixed results can be attributed at least in part to differences in the strengths of the baseline resolvers employed in the evaluation: the stronger the baseline is, the harder it would be to improve its performance. Since different researchers employed different baselines and evaluated their resolvers on different feature sets, it is not easy to draw general conclusions on the usefulness of different kinds of semantic features. To facilitate comparison of the usefulness of different kinds of semantic features, we believe that it is worthwhile to re-evaluate them using the standard evaluation setup provided by the CoNLL-2011 and 2012 shared tasks.

VI. Conclusion

We presented an overview of the models and features developed for learning-based entity coreference resolution in the past two decades, as well as the corpora and metrics used in the evaluation of these computational models. Despite the continued progress on this task, it is far from being solved: the best CoNLL scores reported to date on the CoNLL-2012 official evaluation data for English and Chinese are 65.29 and 63.66 respectively [39]. Recent results suggest that the performance of coreference models that do not employ sophisticated knowledge is plateauing [38]. Hence, one of the fruitful avenues of future research will likely come from the incorporation of sophisticated knowledge sources.

Acknowledgment

This work was supported in part by NSF Grant IIS-1219142. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views or official policies of NSF.

References

[1] R. Mitkov, B. Boguraev, and S. Lappin, "Introduction to the special issue on computational anaphora resolution," *Computational Linguistics*, vol. 27, no. 4, pp. 473–477, 2001.

[2] V. Ng and C. Cardie, "Improving machine learning approaches to coreference resolution," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2012, pp. 104–111.

[3] T. Winograd, *Understanding Natural Language*. New York: Academic Press, Inc., 1972.

[4] H. Levesque, "The winograd schema challenge," in *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 2011.

[5] M. Poesio, R. Stuckardt, and Y. V. (Eds.), *Anaphora Resolution: Algorithms, Resources, and Evaluation*. Springer Verlag, 2016.

[6] MUC-6, *Proceedings of the Sixth Message Understanding Conference*. San Francisco, CA: Morgan Kaufmann, 1995.

[7] MUC-7, *Proceedings of the Seventh Message Understanding Conference*. San Francisco, CA: Morgan Kaufmann, 1998.

[8] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman, "A model-theoretic coreference scoring scheme," in *Proceedings of the Sixth Message Understanding Conference*, 1995, pp. 45–52.

[9] A. Bagga and B. Baldwin, "Algorithms for scoring coreference chains," in *Proceedings of the Linguistic Coreference Workshop at The First International Conference on Language Resources and Evaluation*, 1998, pp. 563–566.

[10] X. Luo, "On coreference resolution performance metrics," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005, pp. 25–32.

[11] A. McCallum and B. Wellner, "Conditional models of identity uncertainty with application to noun coreference," in *Advances in Neural Information Proceeing Systems*, 2004.

[12] V. Stoyanov, N. Gilbert, C. Cardie, and E. Riloff, "Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 2009, pp. 656–664.

[13] S. Pradhan, L. Ramshaw, M. Marcus, M. Palmer, R. Weischedel, and N. Xue, "CoNLL-2011 Shared Task: Modeling unrestricted coreference in Ontonotes," in *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, 2011, pp. 1–27.

[14] S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang, "CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning - Shared Task*, 2012, pp. 1–40.

[15] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, "OntoNotes: The 90% solution," in *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, 2006, pp. 57–60.

[16] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[17] M. Recasens and E. Hovy, "BLANC: Implementing the Rand Index for coreference evaluation," *Natural Language Engineering*, vol. 17, no. 4, pp. 485–510, 2011.

[18] W. M. Soon, H. T. Ng, and D. C. Y. Lim, "A machine learning approach to coreference resolution of noun phrases," *Computational Linguistics*, vol. 27, no. 4, pp. 521–544, 2001.

[19] V. Ng and C. Cardie, "Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution," in *Proceedings of the 19th International Conference on Computational Linguistics*, 2002, pp. 730–736.

[20] P. Denis and J. Baldridge, "Global, joint determination of anaphoricity and coreference resolution using integer programming," in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, 2007, pp. 236–243.

[21] R. Iida, K. Inui, H. Takamura, and Y. Matsumoto, "Incorporating contextual cues in trainable models for coreference resolution," in *Proceedings of the EACL Workshop on The Computational Treatment of Anaphora*, 2003.

[22] X. Yang, G. Zhou, J. Su, and C. L. Tan, "Coreference resolution using competitive learning approach," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 2003, pp. 176–183.

[23] P. Denis and J. Baldridge, "Specialized models and ranking for coreference resolution," in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 2008, pp. 660–669.

[24] A. Rahman and V. Ng, "Supervised models for coreference resolution," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 2009, pp. 968–977.

[25] X. Luo, A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos, "A mention-synchronous coreference resolution algorithm based on the Bell tree," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, 2004, pp. 135–142.

[26] X. Yang, J. Su, G. Zhou, and C. L. Tan, "An NP-cluster based approach to coreference resolution," in *Proceedings of the 20th International Conference on Computational Linguistics*, 2004.

[27] H. Lee, A. Chang, Y. Peirsman, N. Chambers, M. Surdeanu, and D. Jurafsky, "Deterministic coreference resolution based on entity-centric, precision-ranked rules," *Computational Linguistics*, vol. 39, no. 4, pp. 885–916, 2013.

[28] L. Ratinov and D. Roth, "Learning-based multi-sieve co-reference resolution with knowledge," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 1234–1244.

[29] A. Culotta, M. Wick, and A. McCallum, "First-order probabilistic models for coreference resolution," in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, 2007, pp. 81–88.

[30] V. Stoyanov and J. Eisner, "Easy-first coreference resolution," in *Proceedings of the 24th International Conference on Computational Linguistics*, 2012, pp. 2519–2534.

[31] T. Finley and T. Joachims, "Supervised clustering with support vector machines," in *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

[32] E. Fernandes, C. dos Santos, and R. Milidiú, "Latent structure perceptron with feature induction for unrestricted coreference resolution," in *Joint Conference on EMNLP and CoNLL - Shared Task*, 2012, pp. 41–48.

[33] J. Chu, Y and T. H. Liu, "On the shortest arborescence of a directed graph," *Science Sinica*, vol. 14, no. 1, pp. 1396–1400, 1965.

[34] J. Edmonds, "Optimum branchings," *Journal of Research of the National Bureau of Standards*, vol. 71B, pp. 233–240, 1967.

[35] G. Durrett and D. Klein, "Easy victories and uphill battles in coreference resolution," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1971–1982.

[36] S. Wiseman, A. M. Rush, S. Shieber, and J. Weston, "Learning anaphoricity and antecedent ranking features for coreference resolution," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 1416–1426.

[37] A. Björkelund and J. Kuhn, "Learning structured perceptrons for coreference resolution with latent antecedents and non-local features," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 47–57.

[38] S. Wiseman, A. M. Rush, and S. M. Shieber, "Learning global features for coreference resolution," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 994–1004.

[39] K. Clark and C. D. Manning, "Improving coreference resolution by learning entity-level distributed representations," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 643–653.

[40] I. Dagan and A. Itai, "Automatic processing of large corpora for the resolution of anaphora references," in *Proceedings of the 13th International Conference on Computational Linguistics*, 1990, pp. 330–332.

[41] A. Kehler, D. Appelt, L. Taylor, and A. Simma, "The (non)utility of predicate-argument frequencies for pronoun interpretation," in *Human Language Technology Conference and Conference of the North American Chapter of the Association for Computational Linguistics: Main Proceedings*, 2004, pp. 289–296.

[42] X. Yang, J. Su, and C. L. Tan, "Improving pronoun resolution using statistics-based semantic compatibility information," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, 2005, pp. 165–172.

[43] S. P. Ponzetto and M. Strube, "Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution," in *Proceedings of the Human Language Technology Conference and Conference of the North American Chapter of the Association for Computational Linguistics*, 2006, pp. 192–199.

[44] M. Bansal and D. Klein, "Coreference semantics from web features," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2012, pp. 389–398.

[45] H. Hajishirzi, L. Zilles, D. S. Weld, and L. Zettlemoyer, "Joint coreference resolution and named-entity linking with multi-pass sieves," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 289–299.

[46] A. Rahman and V. Ng, "Coreference resolution with world knowledge," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 814–824.

[47] M. Hearst, "Automatic acquisition of hyponyms from large text corpora," in *Proceedings of the Fourteenth International Conference on Computational Linguistics*, 1992.

[48] E. Bengtson and D. Roth, "Understanding the values of features for coreference resolution," in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 2008, pp. 294–303.

[49] X. Yang and J. Su, "Coreference resolution using semantic relatedness information from automatically discovered patterns," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, 2007, pp. 528–535.