Machine Learning for Entity Coreference Resolution: A Retrospective Look at Two Decades of Research

Vincent Ng
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688
vince@.hlt.utdallas.edu

Abstract

Though extensively investigated since the 1960s, entity coreference resolution, a core task in natural language understanding, is far from being solved. 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 and discusses a hard entity coreference task, the Winograd Schema Challenge, which has recently received a lot of attention in the AI community.

1 Introduction

Entity coreference resolution is the task of determining which entity mentions in a text or dialogue refer to the same real-world entity. Despite being actively investigated for 50 years in the natural language processing (NLP) community, it is still far from being solved. To better understand the difficulty of the task, 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 stems from its reliance on sophisticated knowledge sources and inference mechanisms (Mitkov et al. 2001). 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 high-level NLP applications, such as question answering, text summarization, and machine translation.

Our goal in this paper is to provide the AI audience 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. (2016). We believe that the entity coreference task will be of interest to the general AI audience. As Levesque (2011) argued, the pronoun resolution task defined in the Winograd Schema Challenge (WSC), which has recently become popular in the AI community, is an appealing alternative to the Turing Test.

2 Brief History and Some Perspectives

Research on coreference resolution exhibited a gradual shift from heuristic approaches to machine learning approaches in the 1990s. Learning-based 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 (1995) and MUC-7 (1998) 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 (Vilain et al. 1995), which was later criticized for its inability to reward successful identification of singleton clusters. Nevertheless, 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 the series 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 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 scoring metrics were developed during this period, both of which aimed to address the aforementioned weakness of the MUC metric. Specifically, B^3 (Bagga and Baldwin 1998), a mention-based metric that was originally developed to evaluate cross-document coreference systems, computes the recall and precision for each mention and then aggregates them into overall recall and precision values, whereas CEAF (Luo 2005) is an entity-based metric that evaluates coreference outputs based on the best alignment between the clusters in the gold partition and those in the system-generated partition.

Direct comparisons among the different coreference systems developed during this period 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 (McCallum and Wellner 2004), 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.

The CoNLL 2011 and 2012 shared tasks. CoNLL 2011 (Pradhan et al. 2011) and 2012 (Pradhan et al. 2012) focused on English and multilingual (English, Chinese, and Arabic) coreference resolution, respectively, using the OntoNotes 5.0 corpus (Hovy et al. 2006) 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 (Pradhan et al. 2011), which is the unweighted average of MUC, B^3, and CEAF. Virtually all resolvers developed since 2011 were evaluated on this corpus.

3 Models

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

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* (Soon et al. 2001) and *best-first* clustering (Ng and Cardie 2002b), is needed to coordinate the pairwise decisions and construct a partition.

It was around this time that Ng and Cardie (2002a) 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).[1]

Interest in anaphoricity determination is stimulated primarily by the fact that proper modeling of anaphoricity could substantially simplify the coreference task. Specifically, a coreference model will only need to resolve mentions that are determined to be anaphoric by the anaphoricity model. However, inaccurate anaphoricity determination

[1]Recasens et al. (2013) have recently proposed a closely related task that involves determining whether a mention is a singleton or is part of a coreference chain.

can hurt coreference performance, as errors may propagate from the anaphoricity component to the coreference component. Given that anaphoricity determination is by no means an easier task than coreference resolution, some early approaches choose not to explicitly model anaphoricity, implicitly positing a mention as non-anaphoric if no antecedent is selected for it by the clustering algorithm.

Mention-Ranking Models

Recasting coreference as a classification task may not be a good idea, however. Recall that mention-pair models 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 candidate antecedents of a mention to be ranked *simultaneously* (Iida et al. 2003; Yang et al. 2003; Denis and Baldridge 2008). Since a mention ranker simply imposes a ranking on candidate antecedents, it cannot determine whether a mention is anaphoric. A natural way to address this problem is to apply an independently trained anaphoricity classifier to identify non-anaphoric mentions prior to ranking.

Entity-Based Models

Mention-pair models have limitations in their *expressiveness*: they can only employ *local* features (i.e., 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* (Luo et al. 2004; Yang et al. 2004). 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 (Rahman and Ng 2009). They rank preceding clusters rather than candidate antecedents, and have been shown to outperform entity-mention models and the mention-based models.

Culotta et al. (2007) and Stoyanov and Eisner (2012) train coreference models to perform *agglomerative* clustering. Initially, each mention is in its own cluster. In each iteration, their models, which are learned using online learners, select the two "best" clusters to merge. Hence, these models can exploit cluster-level features.

Daumé III and Marcu (2005a) train a model that searches the Bell tree.[2] Informally, a node in the ith level of a Bell tree corresponds to an ith-order partial partition (i.e., a partition of the first i mentions of the given document), and the ith level of the tree contains all possible ith-order partial partitions. Hence, a leaf node contains a complete partition of the mentions. The goal is to search for the leaf node that contains the most probable partition. The search starts at the root, and a partitioning of the mentions is incrementally constructed as we move down the tree. Specifically, based on the coreference decisions it has made in the first $i-1$ levels of the tree, the model determines at the ith level whether the ith mention should start a new cluster, or to which preceding cluster it should be assigned. Precisely *how* the model searches the tree (i.e., the search strategy) is learned using the Learning as Search Optimization framework (Daumé III and Marcu 2005b). The model is entity-based as it can exploit cluster-level features computed based on the clusters in the partial partition constructed so far in the search process.

Partition-Based Models

Taking the idea of modeling entities a step further, one can train models that directly induce a coreference partition on a set of mentions. McCallum and Wellner (2004), for instance, 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 (2005), 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 for this is that inference in such models with arbitrary cluster-level features is intractable. As a result, both McCallum and Wellner (2004) and Finley and Joachims (2005) resort to using only local features when training their models. Another reason is

[2]More precisely, Daumé and Marcu perform joint entity detection and coreference resolution, so their search space is more complicated than that defined by the Bell tree, but for ease of exposition, we describe without loss of generality how their method works by assuming that the state space is defined by a Bell tree.

that they force us to classify each pair of mentions, which is not desirable as not all links are equally easy to identify.

Graph-Based Approaches

Several attempts have been made to cast coreference as a (hyper)graph partitioning problem. Given a test document, a (hyper)graph is first constructed, where the nodes and (hyper)edges typically correspond to the mentions and their compatibility, respectively. (Hyper)edge weights can be computed using a learned mention-pair model (Nicolae and Nicolae 2006), via collecting simple statistics from the training data (Cai and Strube 2010; Sapena et al. 2013), or learned to maximize an objective function (McCallum and Wellner 2004). A (hyper)graph partitioning algorithm can then be applied to obtain coreference clusters. For instance, spectral clustering, correlation clustering, and relaxation labeling are used by Cai and Strube (2010), McCallum and Wellner (2004), and Sapena et al. (2013) respectively. Note that hyperedges (as opposed to edges) connect multiple nodes and therefore enable the use of cluster-level features.

Joint Models

The successes of joint models developed for various NLP tasks has motivated their application to coreference resolution. By making classification decisions jointly, joint models enable the incorporation of relational background knowledge that encodes task-specific consistency constraints.

One such early attempt was made by Denis and Baldridge (2007), who perform joint inference for anaphoricity determination and coreference resolution using Integer Linear Programming (ILP) (Roth and Yih 2004). They exploit the constraint that a mention should be classified as anaphoric if and only if the mention-pair coreference model finds an antecedent for it. The classification confidence values provided by an anaphoricity classifier and a mention-pair model are used as prior knowledge in the joint inference process.

Klenner (2007) and Finkel and Manning (2008) enforce transitivity by performing ILP-based joint inference over pairwise coreference decisions. The classification confidence of each pair of mentions according to a mention-pair model is employed as prior knowledge for joint inference.

Unlike the aforementioned joint models where classifiers are first trained to provide prior knowledge for joint inference, Song et al. (2012) combine pairwise classification and clustering using a Markov Logic Network (MLN) (Domingos and Lowd 2009). Specifically, they encode features commonly-used in mention-pair models as soft formulas and transitivity as hard formulas in the MLN, learn the weights of the soft formulas, and use the resulting MLN to jointly perform classification and clustering.

Poon and Domingos (2008) make one of the few attempts to perform joint inference for *unsupervised* coreference resolution using MLNs. Their MLN is a cluster-based model, which assigns each mention to a coreference cluster. They exploit standard constraints on coreference, such as agreement on gender, number, and semantic class.

Rather than performing joint inference, Rahman and Ng (2009) perform joint *learning* for anaphoricity determination and coreference resolution when training their ranking models. The idea is to augment the set of candidate antecedents with a *null* candidate that receives the highest rank if and only if the mention to be resolved is non-anaphoric.

Semi-Supervised and Unsupervised Models

Semi-supervised and unsupervised models aim to reduce or even eliminate a model's reliance on annotated data. The early 2000s have seen the application of semi-supervised learners such as co-training and self-training to pronoun and coreference resolution, where resolution models are bootstrapped from a small amount of labeled data (Müller et al. 2002; Ng and Cardie 2003; Kehler et al. 2004a). Nevertheless, much work in this area has focused on intelligently designing probabilistic generative models for unsupervised coreference resolution (Haghighi and Klein 2007; 2010; Ng 2008) and pronoun resolution (Bergsma and Cherry 2005; Charniak and Elsner 2009). While some unsupervised models have rivaled their supervised counterparts in performance, the large amount of training data provided by OntoNotes has significantly improved the performance of supervised models in recent years. In fact, it is possible to achieve state-of-the-art performance by training supervised coreference models using only lexical features (Björkelund and Nugues 2011; Durrett and Klein 2013).

Easy-First Models

Easy-first coreference models aim to make easy linking decisions first. Like entity-based models, easy-first models can employ cluster-level features: the information extracted from the clusters established thus far can be used to help identify the difficult links.

The most well-known resolver that employs an easy-first approach is arguably Stanford's resolver (Lee et al. 2011), which won the CoNLL-2011 shared task. This resolver is composed of 12 sieves, each of which is composed of a set of hand-crafted rules for classifying a *subset* of the mention pairs in the test set. Being an easy-first approach, the sieves are arranged as a pipeline in decreasing order of precision. While later sieves can exploit the decisions made by earlier sieves, these earlier decisions cannot be overridden even if they are erroneous.

Ratinov and Roth (2012) attempt to improve Stanford's architecture. In addition to employing different sieves, their easy-first sieve-based architecture differs from Stanford's in two major aspects. First, each sieve is associated with a *learned* classifier rather than a set of hand-crafted rules. Second, they allow earlier decisions to be overridden by later sieves. Specifically, when training the classifier for a given sieve, the classifier (1) employs features that encode the predictions of all the previous $i-1$ sieves and then (2) decides whether an earlier decision should be overridden or not.

Sieve-based models are important because (1) they are extensible (new sieves can be easily added) and allow a task as complex as coreference resolution to be decomposed into smaller, more manageable tasks; (2) their easy-first nature may be the key to make entity-mention models work; and (3) the rule-based implementation of these models can be a promising approach to coreference resolution for languages for which coreference-annotated data is not available.

Tree-Based Models

As mentioned before, not all coreference links are equally easy to identify. Fortunately, to establish a cluster of n mentions, only $n-1$ links are needed. So, rather than learning a partition, Fernandes et al. (2012) (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 (Chu and Liu 1965; Edmonds 1967). 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 entropy-guided feature induction method learns feature conjunctions, which are derived from the paths of a decision tree-based mention-pair model.

Antecedent Structure-Based Models

Durrett and Klein (2013) (D&K) propose training an antecedent structure-based model for coreference resolution. Their model 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.

D&K's model belongs to a recently popular line of work that views coreference resolution as a structured prediction task. Martschat and Strube (2015) show that several commonly-used coreference models (mention-pair models, mention-ranking models, and tree-based models) can in fact be viewed as predicting different latent structures, and propose a unified framework in which these models are trained to predict their respective structures using a latent structured perceptron learning algorithm. This unified framework could help us directly compare different models by identifying their relative strengths and weaknesses.

Neural Models

Like D&K, Wiseman et al. (2015) train a mention-ranking model that employs a task-specific loss function. However, rather than following the recent trend on training *linear* models using feature conjunctions (e.g., Fernandes et al. (2012), Durrett and Klein (2013), Björkelund and Kuhn (2014)), some of which are rather complex, Wiseman et al. pioneered employing 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. (2016) and Clark and Manning (2016) further improved the performance of neural coreference models by incorporating entity-based features. These are the first attempts to learn non-linear models of coreference resolution. Given their promising results, they deserve further investigations.

4 Semantics and World Knowledge

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 sophisticated features for coreference resolution, as described below.

Selectional preference is one of the earliest kinds of semantic knowledge exploited for coreference resolution (Dagan and Itai 1990; Kehler et al. 2004b; Yang et al. 2005). 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 (Soon et al. 2001; Ponzetto and Strube 2006) or clusters induced from the Google n-gram corpus (Bansal and Klein 2012).

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 (2012) extract from Wikipedia its Wiki category, gender, and nationality, and Hajishirzi et al. (2013) 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 (Rahman and Ng 2011): 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 (2012) employ a context-sensitive entity disambiguation system, while Hajishirzi et al. (2013) 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 (Hearst 1992). 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 (2008) have employed as features the generic semantic relations (e.g., synonymy, hypernymy, antonymy) extracted from WordNet for two common nouns. Hearst (1992) has proposed other lexico-syntactic patterns that capture different lexical semantic relations between nouns. Yang and Su (2007) 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 (2006) to encode features based on various measures of WordNet similarity, which have been shown to improve their baseline system.

PropBank-style *semantic roles* (e.g., ARG0, ARG1) have also been used for coreference resolution (Ponzetto and Strube 2006). 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 (2006) resolver, 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 (2011) 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. This way of capturing event relatedness, however, is still crude. In light of this problem, Rahman and Ng (2012) capture event relatedness using the *narrative chains* learned by Chambers and Jurafsky (2009), whereas Bean and Riloff (2004) learn domain-specific narrative chains by bootstrapping from a small set of coreferent noun pairs.

Rahman and Ng (2012) and Peng et al. (2015) examine difficult-to-resolve pronouns in the WSC. The WSC was motivated by the following pair of sentences, which was originally used by Winograd (1972) to illustrate the difficulty of natural language understanding:

(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 typically requires background knowledge. Levesque (2011) argued that the resolution of difficult-to-resolve pronouns in *twin* sentences like these, which he refers to as the WSC, constitutes a task that can serve as an appealing alternative to the Turing Test.

To address the WSC, Rahman and Ng (2012) employ a variety of semantic knowledge sources, including narrative chains, FrameNet semantic roles, and sentiment/polarity information. On the other hand, Peng et al. (2015) employ hand-crafted patterns, which they call *Predicate Schemas*, to collect co-occurrence statistics of the two predicates and the discourse connective from a large, unannotated corpus, and exploit the statistics as features for their resolver. The WSC is currently being promoted by Commonsense Reasoning[3], so we expect to see continued progress on this task.

Generally speaking, the results of employing semantic and world knowledge to improve knowledge-poor coreference resolvers are mixed. We believe 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. While a recent evaluation by Durrett and Klein (2013) suggests that incorporating shallow semantic features (e.g., named entity types, WordNet hypernymy) does not improve their state-of-the art mention-ranking model that uses only morpho-syntactic features, a more comprehensive evaluation of existing semantic features is needed. Nevertheless, recent results seem to suggest that the performance of coreference models that do not employ sophisticated knowledge is plateauing (Wiseman et al. 2016). In his invited talk at the NAACL HLT 2016 workshop on Coreference Resolution beyond OntoNotes, Michael Strube conjectured that performance gains beyond the current state of the art will likely come from the incorporation of sophisticated knowledge sources.

5 Concluding Remarks

While researchers are making continued progress on the entity coreference task despite its difficulty, a natural question is: what are the promising directions for future work?

Rather than contemplating entity coreference resolution as a standalone task, it may be worthwhile to investigate **cross-task joint models** that involve entity coreference as one of a set of related tasks to be learned, so that cross-task hard/soft constraints can be enforced to improve model learning. This direction seems promising considering Durrett and Klein's (2014) recent success in joint inference for entity coreference, semantic typing, and entity linking.

If joint modeling is not possible (e.g., because annotated data is not available for training models for the related tasks), it is likely that we need to employ **sophisticated features** to improve state-of-the-art resolvers despite the difficulty in extracting/inducing such features. Recall that Wiseman et al. (2015) obtained promising results by learning non-linear

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

representations from *raw* features. One can take this further and learn such representations from *complex* features, including those that encode the world knowledge extracted from lexical knowledge bases. In addition, despite various attempts to extract world knowledge, almost all related efforts have focused on extracting knowledge about *entities*. As we saw in the example in the introduction, commonsense knowledge that is not centered around an entity (e.g., it does not make sense for Person A to summon Person B to treat Person B's problem) is equally important.

Despite their impressive performance, supervised entity coreference models cannot be applied to the vast majority of the world's **low-resource languages** for which coreference-annotated data is not readily available. It would be interesting to examine whether there are language-specific issues that could affect the effective application of unsupervised, semi-supervised, and annotation projection approaches to coreference resolution involving less-studied languages. In addition, if large lexical knowledge bases do not exist for the target language, it would be important to investigate alternative methods for obtaining world knowledge.

Finally, many tasks in the family of coreference problems are arguably more challenging than the identity coreference task we examined in this paper and deserve more attention in the community. These include **non-identity coreference tasks** such as *bridging* (part-whole relations, set-subset relations) and **event coreference**, which assumes as input the noisy outputs of event extraction and entity coreference.

Acknowledgments

We thank the three anonymous reviewers for their detailed and insightful comments on an earlier draft of the paper. This work was supported in part by NSF Grant IIS-1219142.

References

Bagga, A., and Baldwin, B. 1998. Algorithms for scoring coreference chains. *LREC Linguistic Coreference Workshop*.

Bansal, M., and Klein, D. 2012. Coreference semantics from web features. *ACL*.

Bean, D., and Riloff, E. 2004. Unsupervised learning of contextual role knowledge for coreference resolution. *HLT-NAACL*.

Bengtson, E., and Roth, D. 2008. Understanding the value of features for coreference resolution. *EMNLP*.

Bergsma, S., and Cherry, C. 2005. An Expectation Maximization approach to pronoun resolution. *CoNLL*.

Björkelund, A., and Kuhn, J. 2014. Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. *ACL*.

Björkelund, A., and Nugues, P. 2011. Exploring lexicalized features for coreference resolution. *CoNLL: Shared Task*.

Cai, J., and Strube, M. 2010. End-to-end coreference resolution via hypergraph partitioning. *COLING*.

Chambers, N., and Jurafsky, D. 2009. Unsupervised learning of narrative schemas and their participants. *ACL-IJCNLP*.

Charniak, E., and Elsner, M. 2009. EM works for pronoun anaphora resolution. *EACL*.

Chu, Y, J., and Liu, T. H. 1965. On the shortest arborescence of a directed graph. *Science Sinica* 14(1):1396–1400.

Clark, K., and Manning, C. D. 2016. Improving coreference resolution by learning entity-level distributed representations. *ACL*.

Culotta, A.; Wick, M.; and McCallum, A. 2007. First-order probabilistic models for coreference resolution. *NAACL HLT*.

Dagan, I., and Itai, A. 1990. Automatic processing of large corpora for the resolution of anaphora references. *COLING*.

Daumé III, H., and Marcu, D. 2005a. A large-scale exploration of effective global features for a joint entity detection and tracking model. *HLT/EMNLP*.

Daumé III, H., and Marcu, D. 2005b. Learning as search optimization: Approximate large margin methods for structured prediction. *ICML*.

Denis, P., and Baldridge, J. 2007. Global, joint determination of anaphoricity and coreference resolution using integer programming. *NAACL HLT*.

Denis, P., and Baldridge, J. 2008. Specialized models and ranking for coreference resolution. *EMNLP*.

Domingos, P., and Lowd, D. 2009. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan & Claypool.

Durrett, G., and Klein, D. 2013. Easy victories and uphill battles in coreference resolution. *EMNLP*.

Durrett, G., and Klein, D. 2014. A joint model for entity analysis: Coreference, typing, and linking. *TACL*.

Edmonds, J. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards*.

Fernandes, E.; dos Santos, C.; and Milidiú, R. 2012. Latent structure perceptron with feature induction for unrestricted coreference resolution. *EMNLP-CoNLL: Shared Task*.

Finkel, J. R., and Manning, C. 2008. Enforcing transitivity in coreference resolution. *ACL:HLT Short Papers*.

Finley, T., and Joachims, T. 2005. Supervised clustering with support vector machines. *ICML*.

Haghighi, A., and Klein, D. 2007. Unsupervised coreference resolution in a nonparametric bayesian model. *ACL*.

Haghighi, A., and Klein, D. 2010. Coreference resolution in a modular, entity-centered model. *NAACL HLT*.

Hajishirzi, H.; Zilles, L.; Weld, D. S.; and Zettlemoyer, L. 2013. Joint coreference resolution and named-entity linking with multi-pass sieves. *EMNLP*.

Hearst, M. 1992. Automatic acquisition of hyponyms from large text corpora. *COLING*.

Hovy, E.; Marcus, M.; Palmer, M.; Ramshaw, L.; and Weischedel, R. 2006. Ontonotes: The 90% solution. *HLT-NAACL*.

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

Kehler, A.; Appelt, D.; Taylor, L.; and Simma, A. 2004a. Competitive self-trained pronoun interpretation. *HLT-NAACL Short Papers*.

Kehler, A.; Appelt, D.; Taylor, L.; and Simma, A. 2004b. The (non)utility of predicate-argument frequencies for pronoun interpretation. *HLT-NAACL*.

Klenner, M. 2007. Enforcing consistency on coreference sets. *RANLP*.

Lee, H.; Chang, A.; Peirsman, Y.; Chambers, N.; Surdeanu, M.; and Jurafsky, D. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*.

Levesque, H. 2011. The Winograd Schema Challenge. *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*.

Luo, X.; Ittycheriah, A.; Jing, H.; Kambhatla, N.; and Roukos, S. 2004. A mention-synchronous coreference resolution algorithm based on the Bell tree. *ACL*.

Luo, X. 2005. On coreference resolution performance metrics. *HLT/EMNLP*.

Martschat, S., and Strube, M. 2015. Latent structures for coreference resolution. *TACL*.

McCallum, A., and Wellner, B. 2004. Conditional models of identity uncertainty with application to noun coreference. *NIPS*.

Mitkov, R.; Boguraev, B.; and Lappin, S. 2001. Introduction to the special issue on computational anaphora resolution. *Computational Linguistics*.

MUC-6. 1995. *Proceedings of MUC-6*.

MUC-7. 1998. *Proceedings of MUC-7*.

Müller, C.; Rapp, S.; and Strube, M. 2002. Applying co-training to reference resolution. *ACL*.

Ng, V., and Cardie, C. 2002a. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. *COLING*.

Ng, V., and Cardie, C. 2002b. Improving machine learning approaches to coreference resolution. *ACL*.

Ng, V., and Cardie, C. 2003. Weakly supervised natural language learning without redundant views. *HLT-NAACL*.

Ng, V. 2008. Unsupervised models for coreference resolution. *EMNLP*.

Nicolae, C., and Nicolae, G. 2006. BestCut: A graph algorithm for coreference resolution. *EMNLP*.

Peng, H.; Khashabi, D.; and Roth, D. 2015. Solving hard coreference problems. *NAACL HLT*.

Poesio, M.; Stuckardt, R.; and Versley, Y. (Eds.). 2016. *Anaphora Resolution: Algorithms, Resources, and Evaluation*. Springer Verlag.

Ponzetto, S. P., and Strube, M. 2006. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. *HLT-NAACL*.

Poon, H., and Domingos, P. 2008. Joint unsupervised coreference resolution with Markov Logic. *EMNLP*.

Pradhan, S.; Ramshaw, L.; Marcus, M.; Palmer, M.; Weischedel, R.; and Xue, N. 2011. CoNLL-2011 Shared Task: Modeling unrestricted coreference in OntoNotes. *CoNLL: Shared Task*.

Pradhan, S.; Moschitti, A.; Xue, N.; Uryupina, O.; and Zhang, Y. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. *EMNLP-CoNLL: Shared Task*.

Rahman, A., and Ng, V. 2009. Supervised models for coreference resolution. *EMNLP*.

Rahman, A., and Ng, V. 2011. Coreference resolution with world knowledge. *ACL-HLT*.

Rahman, A., and Ng, V. 2012. Resolving complex cases of definite pronouns: The Winograd Schema Challenge. *EMNLP-CoNLL*.

Ratinov, L., and Roth, D. 2012. Learning-based multi-sieve co-reference resolution with knowledge. *EMNLP-CoNLL*.

Recasens, M.; de Marneffe, M.-C.; and Potts, C. 2013. The life and death of discourse entities: Identifying singleton mentions. *NAACL HLT Short Papers*.

Roth, D., and Yih, W.-T. 2004. A linear programming formulation for global inference in natural language tasks. *CoNLL*.

Sapena, E.; Padró, L.; and Turmo J. 2013. A constraint-based hypergraph partitioning approach to coreference resolution. *Computational Linguistics*.

Song, Y.; Jiang, J.; Zhao, W. X.; Li, S.; and Wang, H. 2012. Joint learning for coreference resolution with Markov logic. *EMNLP-CoNLL*.

Soon, W. M.; Ng, H. T.; and Lim, D. C. Y. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*.

Stoyanov, V., and Eisner, J. 2012. Easy-first coreference resolution. *COLING*.

Vilain, M.; Burger, J.; Aberdeen, J.; Connolly, D.; and Hirschman, L. 1995. A model-theoretic coreference scoring scheme. *MUC-6*.

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

Wiseman, S.; Rush, A. M.; Shieber, S.; and Weston, J. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. *ACL-IJCNLP*.

Wiseman, S.; Rush, A. M.; and Shieber, S. M. 2016. Learning global features for coreference resolution. *NAACL HLT*.

Yang, X., and Su, J. 2007. Coreference resolution using semantic relatedness information from automatically discovered patterns. *ACL*.

Yang, X.; Su, J.; and Tan, C. L. 2005. Improving pronoun resolution using statistics-based semantic compatibility information. *ACL*.

Yang, X.; Su, J.; Zhou, G.; and Tan, C. L. 2004. An NP-cluster based approach to coreference resolution. *COLING*.

Yang, X.; Zhou, G.; Su, J.; and Tan, C. L. 2003. Coreference resolution using competition learning approach. *ACL*.