Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:
Human Language Technologies (ACL-HLT), Portland, OR, June 2011, pp. 814--824.

Coreference Resolution with World Knowledge

Altaf Rahman and **Vincent Ng**
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688
{altaf,vince}@hlt.utdallas.edu

Abstract

While world knowledge has been shown to improve learning-based coreference resolvers, the improvements were typically obtained by incorporating world knowledge into a fairly weak baseline resolver. Hence, it is not clear whether these benefits can carry over to a stronger baseline. Moreover, since there has been no attempt to apply different sources of world knowledge in combination to coreference resolution, it is not clear whether they offer complementary benefits to a resolver. We systematically compare commonly-used and under-investigated sources of world knowledge for coreference resolution by applying them to two learning-based coreference models and evaluating them on documents annotated with two different annotation schemes.

1 Introduction

Noun phrase (NP) coreference resolution is the task of determining which NPs in a text or dialogue refer to the same real-world entity. The difficulty of the task stems in part from its reliance on world knowledge (Charniak, 1972). To exemplify, consider the following text segment.

Martha Stewart is hoping people don't run out on her. The celebrity indicted on charges stemming from . . .

Having the (world) knowledge that *Martha Stewart* is a *celebrity* would be helpful for establishing the coreference relation between the two NPs. One may argue that employing heuristics such as subject preference or syntactic parallelism (which prefers resolving an NP to a candidate antecedent that has the same grammatical role) in this example would also allow us to correctly resolve *the celebrity* (Mitkov, 2002), thereby obviating the need for world knowledge. However, since these heuristics are not perfect, complementing them with world knowledge would be an important step towards bringing coreference systems to the next level of performance.

Despite the usefulness of world knowledge for coreference resolution, early learning-based coreference resolvers have relied mostly on morpho-syntactic features (e.g., Soon et al. (2001), Ng and Cardie (2002), Yang et al. (2003)). With recent advances in lexical semantics research and the development of large-scale knowledge bases, researchers have begun to employ world knowledge for coreference resolution. World knowledge is extracted primarily from three data sources, online encyclopedia (e.g., Ponzetto and Strube (2006), Uryupina et al. (2011)), unannotated data (e.g., Daumé III and Marcu (2005), Ng (2007)), and coreference-annotated data (e.g., Bengtson and Roth (2008)).

While each of these three sources of world knowledge has been shown to improve coreference resolution, the improvements were typically obtained by incorporating world knowledge (as features) into a baseline resolver composed of a rather weak coreference model (i.e., the mention-pair model) and a small set of features (i.e., the 12 features adopted by Soon et al.'s (2001) knowledge-lean approach). As a result, some questions naturally arise. First, can world knowledge still offer benefits when used in combination with a richer set of features? Second, since automatically extracted world knowledge is typically noisy (Ponzetto and Poesio, 2009), are recently-developed coreference models more noise-tolerant than the mention-pair model, and if so, can they profit more from the noisily extracted world knowledge? Finally, while different world knowledge sources have been shown to be useful when applied in isolation to a coreference system, do they offer complementary benefits and therefore can further improve a resolver when applied in combination?

We seek answers to these questions by conducting a systematic evaluation of different world knowledge sources for learning-based coreference resolution. Specifically, we (1) derive world knowledge from encyclopedic sources that are under-investigated for coreference resolution, including FrameNet (Baker et al., 1998) and YAGO (Suchanek et al., 2007), in addition to coreference-annotated data and unannotated data; (2) incorporate such knowledge as features into a richer baseline feature set that we previously employed (Rahman and Ng, 2009); and (3) evaluate their utility using two coreference models, the traditional mention-pair model (Soon et al., 2001) and the recently developed cluster-ranking model (Rahman and Ng, 2009).

Our evaluation corpus contains 410 documents, which are coreference-annotated using the ACE annotation scheme as well as the OntoNotes annotation scheme (Hovy et al., 2006). By evaluating on two sets of coreference annotations for the same set of documents, we can determine whether the usefulness of world knowledge sources for coreference resolution is dependent on the underlying annotation scheme used to annotate the documents.

2 Preliminaries

In this section, we describe the corpus, the NP extraction methods, the coreference models, and the evaluation measures we will use in our evaluation.

2.1 Data Set

We evaluate on documents that are coreference-annotated using both the ACE annotation scheme and the OntoNotes annotation scheme, so that we can examine whether the usefulness of our world knowledge sources is dependent on the underlying coreference annotation scheme. Specifically, our data set is composed of the 410 English newswire articles that appear in both OntoNotes-2 and ACE 2004/2005. We partition the documents into a training set and a test set following a 80/20 ratio.

ACE and OntoNotes employ different guidelines to annotate coreference chains. A major difference between the two annotation schemes is that ACE only concerns establishing coreference chains among NPs that belong to the ACE entity types, whereas OntoNotes does not have this restriction. Hence, the OntoNotes annotation scheme should produce more coreference chains (i.e., non-singleton coreference clusters) than the ACE annotation scheme for a given set of documents. For our data set, the OntoNotes scheme yielded 4500 chains, whereas the ACE scheme yielded only 3637 chains.

Another difference between the two annotation schemes is that singleton clusters are annotated in ACE but not OntoNotes. As discussed below, the presence of singleton clusters may have an impact on NP extraction and coreference evaluation.

2.2 NP Extraction

Following common practice, we employ different methods to extract NPs from the documents annotated with the two annotation schemes.

To extract NPs from the ACE-annotated documents, we train a mention extractor on the training texts (see Section 5.1 of Rahman and Ng (2009) for details), which recalls 83.6% of the NPs in the test set. On the other hand, to extract NPs from the OntoNotes-annotated documents, the same method should not be applied. To see the reason, recall that only the NPs in non-singleton clusters are annotated in these documents. Training a mention extractor on these NPs implies that we are learning to extract *non-singleton NPs*, which are typically much smaller in number than the entire set of NPs. In other words, doing so could substantially simplify the coreference task. Consequently, we follow the approach adopted by traditional learning-based resolvers and employ an NP chunker to extract NPs. Specifically, we use the markable identification system in the Reconcile resolver (Stoyanov et al., 2010) to extract NPs from the training and test texts. This identifier recalls 77.4% of the NPs in the test set.

2.3 Coreference Models

We evaluate the utility of world knowledge using the mention-pair model and the cluster-ranking model.

2.3.1 Mention-Pair Model

The mention-pair (MP) model is a classifier that determines whether two NPs are coreferent or not.

Each instance $\boldsymbol{i}(\text{NP}_j,\ \text{NP}_k)$ corresponds to NP_j and NP_k, and is represented by a Baseline feature set consisting of 39 features. Linguistically, these features can be divided into four categories: string-matching, grammatical, semantic, and positional. These features can also be categorized based on whether they are relational or not. Relational features capture the relationship between NP_j and NP_k, whereas non-relational features capture the linguistic property of one of these two NPs. Since space limitations preclude a description of these features, we refer the reader to Rahman and Ng (2009) for details.

We follow Soon et al.'s (2001) method for creating training instances: we create (1) a positive instance for each anaphoric NP, NP_k, and its closest antecedent, NP_j; and (2) a negative instance for NP_k paired with each of the intervening NPs, NP_{j+1}, NP_{j+2}, ..., NP_{k-1}. The classification of a training instance is either positive or negative, depending on whether the two NPs are coreferent in the associated text. To train the MP model, we use the SVM learning algorithm from SVM^{light} (Joachims, 2002).[1]

After training, the classifier is used to identify an antecedent for an NP in a test text. Specifically, each NP, NP_k, is compared in turn to each preceding NP, NP_j, from right to left, and NP_j is selected as its antecedent if the pair is classified as coreferent. The process terminates as soon as an antecedent is found for NP_k or the beginning of the text is reached.

Despite its popularity, the MP model has two major weaknesses. First, since each candidate antecedent for an NP to be resolved (henceforth an *active NP*) is considered independently of the others, this model only determines how good a candidate antecedent is relative to the active NP, but not how good a candidate antecedent is relative to other candidates. So, it fails to answer the critical question of which candidate antecedent is most probable. Second, it has limitations in its expressiveness: the information extracted from the two NPs alone may not be sufficient for making a coreference decision.

2.3.2 Cluster-Ranking Model

The cluster-ranking (CR) model addresses the two weaknesses of the MP model by combining the strengths of the *entity-mention* model (e.g., Luo et al. (2004), Yang et al. (2008)) and the *mention-ranking* model (e.g., Denis and Baldridge (2008)). Specifically, the CR model ranks the preceding clusters for an active NP so that the highest-ranked cluster is the one to which the active NP should be linked. Employing a ranker addresses the first weakness, as a ranker allows all candidates to be compared *simultaneously*. Considering preceding clusters rather than antecedents as candidates addresses the second weakness, as *cluster-level* features (i.e., features that are defined over any subset of NPs in a preceding cluster) can be employed. Details of the CR model can be found in Rahman and Ng (2009).

Since the CR model ranks preceding clusters, a training instance $\boldsymbol{i}(c_j,\ \text{NP}_k)$ represents a preceding cluster, c_j, and an anaphoric NP, NP_k. Each instance consists of features that are computed based solely on NP_k as well as cluster-level features, which describe the relationship between c_j and NP_k. Motivated in part by Culotta et al. (2007), we create cluster-level features from the *relational* features in our feature set using four predicates: NONE, MOST-FALSE, MOST-TRUE, and ALL. Specifically, for each relational feature X, we first convert X into an equivalent set of binary-valued features if it is multi-valued. Then, for each resulting binary-valued feature X_b, we create four binary-valued cluster-level features: (1) NONE-X_b is true when X_b is false between NP_k and each NP in c_j; (2) MOST-FALSE-X_b is true when X_b is true between NP_k and less than half (but at least one) of the NPs in c_j; (3) MOST-TRUE-X_b is true when X_b is true between NP_k and at least half (but not all) of the NPs in c_j; and (4) ALL-X_b is true when X_b is true between NP_k and each NP in c_j.

We train a cluster ranker to jointly learn anaphoricity determination and coreference resolution using SVM^{light}'s ranker-learning algorithm. Specifically, for each NP, NP_k, we create a training instance between NP_k and *each* preceding cluster c_j using the features described above. Since we are learning a joint model, we need to provide the ranker with the option to start a new cluster by creating an additional training instance that contains the non-relational features describing NP_k. The rank value of a training instance $\boldsymbol{i}(c_j, \text{NP}_k)$ created for NP_k is the rank of c_j among the competing clusters. If NP_k is anaphoric, its rank is HIGH if NP_k belongs to c_j, and LOW otherwise. If NP_k is non-anaphoric, its rank is

[1]For this and subsequent uses of the SVM learner in our experiments, we set all parameters to their default values.

LOW unless it is the additional training instance described above, which has rank HIGH.

After training, the cluster ranker processes the NPs in a test text in a left-to-right manner. For each active NP, NP_k, we create test instances for it by pairing it with each of its preceding clusters. To allow for the possibility that NP_k is non-anaphoric, we create an additional test instance as during training. All these test instances are then presented to the ranker. If the additional test instance is assigned the highest rank value, then we create a new cluster containing NP_k. Otherwise, NP_k is linked to the cluster that has the highest rank. Note that the partial clusters preceding NP_k are formed incrementally based on the predictions of the ranker for the first $k-1$ NPs.

2.4 Evaluation Measures

We employ two commonly-used scoring programs, B^3 (Bagga and Baldwin, 1998) and ϕ_3-CEAF (Luo, 2005), both of which report results in terms of recall (R), precision (P), and F-measure (F) by comparing the gold-standard (i.e., key) partition, KP, against the system-generated (i.e., response) partition, RP.

B^3 computes the recall and precision of each NP and averages these values at the end. Specifically, for each NP, NP_j, B^3 first computes the number of common NPs in KP_j and RP_j, the clusters containing NP_j in KP and RP, respectively, then divides this number by $|KP_j|$ and $|RP_j|$ to obtain the recall and precision of NP_j, respectively. On the other hand, CEAF finds the best one-to-one alignment between the key clusters and the response clusters. Precision and recall are equal to the sum of the number of common NPs between each pair of aligned clusters divided by the total number of NPs in the response and the key, respectively.

A complication arises when B^3 is used to score a response partition containing automatically extracted NPs. Recall that B^3 constructs a mapping between the NPs in the response and those in the key. Hence, if the response is generated using gold-standard NPs, then every NP in the response is mapped to some NP in the key and vice versa. In other words, there are no *twinless* (i.e., unmapped) NPs (Stoyanov et al., 2009). This is not the case when automatically extracted NPs are used, but the original description of B^3 does not specify how twinless NPs should be scored (Bagga and Baldwin, 1998). To address this problem, we set the recall and precision of a twinless NP to zero, regardless of whether the NP appears in the key or the response. On the other hand, CEAF can compare partitions containing twinless NPs without any modification.

Additionally, we remove all the twinless NPs in the response that are singletons. The rationale is simple: since the resolver has successfully identified these NPs as singletons, it should not be penalized, and removing them avoids such penalty.

Since B^3 and CEAF align NPs/clusters, the lack of singleton clusters in the OntoNotes annotations implies that the resulting scores reflect solely how well a resolver identifies coreference links and ignore how well it identifies singleton clusters.

3 Extracting World Knowledge

In this section, we describe how we extract world knowledge for coreference resolution from three different sources: large-scale knowledge bases, coreference-annotated data and unannotated data.

3.1 World Knowledge from Knowledge Bases

We extract world knowledge from two large-scale knowledge bases, YAGO and FrameNet.

3.1.1 Extracting Knowledge from YAGO

We choose to employ YAGO rather than the more popularly-used Wikipedia due to its potentially richer knowledge, which comprises 5 million facts extracted from Wikipedia and WordNet. Each fact is represented as a triple (NP_j, rel, NP_k), where rel is one of the 90 YAGO relation types defined on two NPs, NP_j and NP_k. Motivated in part by previous work (Bryl et al., 2010; Uryupina et al., 2011), we employ the two relation types that we believe are most useful for coreference resolution, TYPE and MEANS. TYPE is essentially an IS-A relation. For instance, the triple (`AlbertEinstein`, TYPE, `physicist`) denotes the fact that *Albert Einstein* is a physicist. MEANS provides different ways of expressing an entity, and therefore allows us to deal with synonymy and ambiguity. For instance, the two triples (`Einstein`, MEANS, `AlbertEinstein`) and (`Einstein`, MEANS, `AlfredEinstein`) denote the facts that *Einstein* may refer to the physicist *Albert Einstein* and the musicologist *Alfred Einstein*, respectively. Hence, the presence of one or both of these relations between two NPs provides strong evidence that the two NPs are coreferent.

YAGO's unification of the information in Wikipedia and WordNet enables it to extract facts that cannot be extracted with Wikipedia or WordNet alone, such as (MarthaStewart, TYPE, celebrity). To better appreciate YAGO's strengths, let us see how this fact was extracted. YAGO first heuristically maps each of the Wiki categories in the Wiki page for *Martha Stewart* to its semantically closest WordNet synset. For instance, the Wiki category AMERICAN TELEVISION PERSONALITIES is mapped to the synset corresponding to sense #2 of the word *personality*. Since *personality* is a direct hyponym of *celebrity* in WordNet, YAGO then extracts the desired fact.

We incorporate the world knowledge from YAGO into our coreference models as a binary-valued feature. If the MP model is used, the YAGO feature for an instance will have the value 1 if and only if the two NPs involved are in a TYPE or MEANS relation. On the other hand, if the CR model is used, the YAGO feature for an instance involving NP_k and preceding cluster c will have the value 1 if and only if NP_k has a TYPE or MEANS relation with any of the NPs in c. Since knowledge extraction from web-based encyclopedia is typically noisy (Ponzetto and Poesio, 2009), we use YAGO to determine whether two NPs have a relation only if one NP is a named entity (NE) of type person, organization, or location according to the Stanford NE recognizer (Finkel et al., 2005) and the other NP is a common noun.

3.1.2 Extracting Knowledge from FrameNet

FrameNet is a lexico-semantic resource focused on semantic frames (Baker et al., 1998). As a schematic representation of a situation, a frame contains the *lexical predicates* that can invoke it as well as the *frame elements* (i.e., semantic roles). For example, the JUDGMENT_COMMUNICATION frame describes situations in which a COMMUNICATOR communicates a judgment of an EVALUEE to an ADDRESSEE. This frame has COMMUNICATOR and EVALUEE as its core frame elements and ADDRESSEE as its non-core frame elements, and can be invoked by more than 40 predicates, such as *acclaim*, *accuse*, *commend*, *decry*, *denounce*, *praise*, and *slam*.

To better understand why FrameNet contains potentially useful knowledge for coreference resolution, consider the following text segment.

Peter Anthony decries program trading as "limiting the game to a few," but he is not sure whether he wants to denounce it because ...

To establish the coreference relation between *it* and *program trading*, it may be helpful to know that *decry* and *denounce* appear in the same frame and the two NPs have the same semantic role.

This example suggests that features encoding both the semantic roles of the two NPs under consideration and whether the associated predicates are "related" to each other in FrameNet (i.e., whether they appear in the same frame) could be useful for identifying coreference relations. Two points regarding our implementation of these features deserve mention. First, since we do not employ verb sense disambiguation, we consider two predicates *related* as long as there is at least one semantic frame in which they both appear. Second, since FrameNet-style semantic role labelers are not publicly available, we use ASSERT (Pradhan et al., 2004), a semantic role labeler that provides PropBank-style semantic roles such as ARG0 (the PROTOAGENT, which is typically the subject of a transitive verb) and ARG1 (the PROTOPATIENT, which is typically its direct object).

Now, assuming that NP_j and NP_k are the arguments of two stemmed predicates, $pred_j$ and $pred_k$, we create 15 features using the knowledge extracted from FrameNet and ASSERT as follows. First, we encode the knowledge extracted from FrameNet as one of three possible values: (1) $pred_j$ and $pred_k$ are in the same frame; (2) they are both predicates in FrameNet but never appear in the same frame; and (3) one or both predicates do not appear in FrameNet. Second, we encode the semantic roles of NP_j and NP_k as one of five possible values: ARG0-ARG0, ARG1-ARG1, ARG0-ARG1, ARG1-ARG0, and OTHERS (the default case).[2] Finally, we create 15 binary-valued features by pairing the 3 possible values extracted from FrameNet and the 5 possible values provided by ASSERT. Since these features are computed over two NPs, we can employ them directly for the MP model. Note that by construction,

[2]We focus primarily on ARG0 and ARG1 because they are the most important core arguments of a predicate and may provide more useful information than other semantic roles.

exactly one of these features will have a non-zero value. For the CR model, we extend their definitions so that they can be computed between an NP, NP_k, and a preceding cluster, c. Specifically, the value of a feature is 1 if and only if its value between NP_k and one of the NPs in c is 1 under its original definition.

The above discussion assumes that the two NPs under consideration serve as predicate arguments. If this assumption fails, we will not create any features based on FrameNet for these two NPs.

To our knowledge, FrameNet has not been exploited for coreference resolution. However, the use of related verbs is similar in spirit to Bean and Riloff's (2004) use of patterns for inducing contextual role knowledge, and the use of semantic roles is also discussed in Ponzetto and Strube (2006).

3.2 World Knowledge from Annotated Data

Since world knowledge is needed for coreference resolution, a human annotator must have employed world knowledge when coreference-annotating a document. We aim to design features that can "recover" such world knowledge from annotated data.

3.2.1 Features Based on Noun Pairs

A natural question is: what kind of world knowledge can we extract from annotated data? We may gather the knowledge that *Barack Obama* is a *U.S. president* if we see these two NPs appearing in the same coreference chain. Equally importantly, we may gather the commonsense knowledge needed for determining *non-coreference*. For instance, we may discover that a *lion* and a *tiger* are unlikely to refer to the same real-world entity after realizing that they never appear in the same chain in a large number of annotated documents. Note that any features computed based on WordNet distance or distributional similarity are likely to incorrectly suggest that *lion* and *tiger* are coreferent, since the two nouns are similar distributionally and according to WordNet.

Given these observations, one may collect the noun pairs from the (coreference-annotated) training data and use them as features to train a resolver. However, for these features to be effective, we need to address *data sparseness*, as many noun pairs in the training data may not appear in the test data.

To improve generalization, we instead create different kinds of *noun-pair-based* features given an annotated text. To begin with, we preprocess each document. A *training* text is preprocessed by randomly replacing 10% of its common nouns with the label UNSEEN. If an NP, NP_k, is replaced with UNSEEN, all NPs that have the same string as NP_k will also be replaced with UNSEEN. A *test* text is preprocessed differently: we simply replace all NPs whose strings are not seen in the training data with UNSEEN. Hence, artificially creating UNSEEN labels from a training text will allow a learner to learn how to handle unseen words in a test text.

Next, we create *noun-pair-based features* for the MP model, which will be used to augment the Baseline feature set. Here, each instance corresponds to two NPs, NP_j and NP_k, and is represented by three groups of *binary-valued* features.

Unseen features are applicable when both NP_j and NP_k are UNSEEN. Either an UNSEEN-SAME feature or an UNSEEN-DIFF feature is created, depending on whether the two NPs are the same string before being replaced with the UNSEEN token.

Lexical features are applicable when neither NP_j nor NP_k is UNSEEN. A lexical feature is an ordered pair consisting of the heads of the NPs. For a pronoun or a common noun, the head is the last word of the NP; for a proper name, the head is the entire NP.

Semi-lexical features aim to improve generalization, and are applicable when neither NP_j nor NP_k is UNSEEN. If exactly one of NP_j and NP_k is tagged as an NE by the Stanford NE recognizer, we create a semi-lexical feature that is identical to the lexical feature described above, except that the NE is replaced with its NE label. On the other hand, if both NPs are NEs, we check whether they are the same string. If so, we create a *NE*-SAME feature, where *NE* is replaced with the corresponding NE label. Otherwise, we check whether they have the same NE tag *and* a word-subset match (i.e., whether the word tokens in one NP appear in the other's list of word tokens). If so, we create a *NE*-SUBSAME feature, where *NE* is replaced with their NE label. Otherwise, we create a feature that is the concatenation of the NE labels of the two NPs.

The noun-pair-based features for the CR model can be generated using essentially the same method. Specifically, since each instance now corresponds to an NP, NP_k, and a preceding cluster, c, we can generate a noun-pair-based feature by applying the above method to NP_k and each of the NPs in c, and its value is the number of times it is applicable to NP_k and c.

3.2.2 Features Based on Verb Pairs

As discussed above, features encoding the semantic roles of two NPs and the relatedness of the associated verbs could be useful for coreference resolution. Rather than encode verb relatedness, we may replace verb relatedness with the verbs themselves in these features, and have the learner learn directly from coreference-annotated data whether two NPs serving as the objects of *decry* and *denounce* are likely to be coreferent or not, for instance.

Specifically, assuming that NP_j and NP_k are the arguments of two stemmed predicates, $pred_j$ and $pred_k$, in the training data, we create five features as follows. First, we encode the semantic roles of NP_j and NP_k as one of five possible values: ARG0-ARG0, ARG1-ARG1, ARG0-ARG1, ARG1-ARG0, and OTHERS (the default case). Second, we create five binary-valued features by pairing each of these five values with the two stemmed predicates. Since these features are computed over two NPs, we can employ them directly for the MP model. Note that by construction, exactly one of these features will have a non-zero value. For the CR model, we extend their definitions so that they can be computed between an NP, NP_k, and a preceding cluster, c. Specifically, the value of a feature is 1 if and only if its value between NP_k and one of the NPs in c is 1 under its original definition.

The above discussion assumes that the two NPs under consideration serve as predicate arguments. If this assumption fails, we will not create any features based on verb pairs for these two NPs.

3.3 World Knowledge from Unannotated Data

Previous work has shown that syntactic appositions, which can be extracted using heuristics from unannotated documents or parse trees, are a useful source of world knowledge for coreference resolution (e.g., Daumé III and Marcu (2005), Ng (2007), Haghighi and Klein (2009)). Each extraction is an NP pair such as <*Barack Obama*, *the president*> and <*Eastern Airlines*, *the carrier*>, where the first NP in the pair is a proper name and the second NP is a common NP. Low-frequency extractions are typically assumed to be noisy and therefore discarded.

We combine the extractions produced by Fleischman et al. (2003) and Ng (2007) to form a database consisting of 1.057 million NP pairs, and create a binary-valued Appositive feature for our models using this database. If the MP model is used, this feature will have the value 1 if and only if the two NPs appear as a pair in the database. On the other hand, if the CR model is used, the feature for an instance involving NP_k and preceding cluster c will have the value 1 if and only if NP_k and at least one of the NPs in c appear as a pair in the database.

4 Evaluation

4.1 Experimental Setup

As described in Section 2, we use as our evaluation corpus the 411 documents that are coreference-annotated using the ACE and OntoNotes annotation schemes. Specifically, we divide these documents into five (disjoint) folds of roughly the same size, training the MP model and the CR model using SVM^{light} on four folds and evaluate their performance on the remaining fold. The features, as well as the NPs used to create the training and test instances, are computed automatically. We employ B^3 and ϕ_3-CEAF to score the output of a resolver.

4.2 Results and Discussion

4.2.1 Baseline Models

Since our goal is to evaluate the effectiveness of the features encoding world knowledge for learning-based coreference resolution, we employ as our baselines the MP model and the CR model trained on the Baseline feature set, which does not contain any features encoding world knowledge. For the MP model, the Baseline feature set consists of the 39 features described in Section 2.3.1; for the CR model, the Baseline feature set consists of the cluster-level features derived from the 39 features used in the Baseline MP model (see Section 2.3.2).

Results of the MP model and the CR model employing the Baseline feature set are shown in rows 1 and 8 of Table 1, respectively. Each row contains the B^3 and CEAF results of the corresponding coreference model when it is evaluated using the ACE and OntoNotes annotations as the gold standard. As we can see, the MP model achieves F-measure scores of 62.4 (B^3) and 60.0 (CEAF) on ACE and 53.3 (B^3)

		ACE						OntoNotes					
		B^3			CEAF			B^3			CEAF		
	Feature Set	R	P	F	R	P	F	R	P	F	R	P	F
	Results for the Mention-Pair Model												
1	Base	56.5	69.7	62.4	54.9	66.3	60.0	50.4	56.7	53.3	48.9	54.5	51.5
2	Base+YAGO Types (YT)	57.3	70.3	63.1	58.7	67.5	**62.8**	51.7	57.9	**54.6**	50.3	55.6	**52.8**
3	Base+YAGO Means (YM)	56.7	70.0	62.7	55.3	66.5	60.4	50.6	57.0	53.6	49.3	54.9	51.9
4	Base+Noun Pairs (WP)	57.5	70.6	**63.4**	55.8	67.4	61.1	51.6	57.6	54.4	49.7	55.4	52.4
5	Base+FrameNet (FN)	56.4	70.9	62.8	54.9	67.5	60.5	50.5	57.5	53.8	48.8	55.1	51.8
6	Base+Verb Pairs (VP)	56.9	71.3	63.3	55.2	67.6	60.8	50.7	57.9	54.0	49.0	55.4	52.0
7	Base+Appositives (AP)	56.9	70.0	62.7	55.6	66.9	60.7	50.3	57.1	53.5	49.1	55.1	51.9
	Results for the Cluster-Ranking Model												
8	Base	61.7	71.2	66.1	59.6	68.8	63.8	53.4	59.2	56.2	51.1	57.3	54.0
9	Base+YAGO Types (YT)	63.5	72.4	67.6	61.7	70.0	**65.5**	54.8	60.6	57.6	52.4	58.9	55.4
10	Base+YAGO Means (YM)	62.0	71.4	66.4	59.9	69.1	64.1	53.9	59.5	56.6	51.4	57.5	54.3
11	Base+Noun Pairs (WP)	64.1	73.4	**68.4**	61.3	70.1	65.4	55.9	62.1	**58.8**	53.5	59.1	**56.2**
12	Base+FrameNet (FN)	61.8	71.9	66.5	59.8	69.3	64.2	53.5	60.0	56.6	51.1	57.9	54.3
13	Base+Verb Pairs (VP)	62.1	72.2	66.8	60.1	69.3	64.4	54.4	60.1	57.1	51.9	58.2	54.9
14	Base+Appositives (AP)	63.1	71.7	67.1	60.5	69.4	64.6	54.1	60.1	56.9	51.9	57.8	54.7

Table 1: Results obtained by applying different types of features in isolation to the Baseline system.

		ACE						OntoNotes					
		B^3			CEAF			B^3			CEAF		
	Feature Set	R	P	F	R	P	F	R	P	F	R	P	F
	Results for the Mention-Pair Model												
1	Base	56.5	69.7	62.4	54.9	66.3	60.0	50.4	56.7	53.3	48.9	54.5	51.5
2	Base+YT	57.3	70.3	63.1	58.7	67.5	62.8	51.7	57.9	54.6	50.3	55.6	52.8
3	Base+YT+YM	57.8	70.9	63.6	59.1	67.9	63.2	52.1	58.3	55.0	50.8	56.0	53.3
4	Base+YT+YM+WP	59.5	71.9	65.1	57.5	69.4	62.9	53.1	59.2	56.0	51.5	57.1	54.1
5	Base+YT+YM+WP+FN	59.6	72.1	65.3	57.2	69.7	62.8	53.1	59.5	56.2	51.3	57.4	54.2
6	Base+YT+YM+WP+FN+VP	59.9	72.5	**65.6**	57.8	70.0	**63.3**	53.4	59.8	**56.4**	51.8	57.7	**54.6**
7	Base+YT+YM+WP+FN+VP+AP	59.7	72.4	65.4	57.6	69.8	63.1	53.2	59.8	56.3	51.5	57.6	54.4
	Results for the Cluster-Ranking Model												
8	Base	61.7	71.2	66.1	59.6	68.8	63.8	53.4	59.2	56.2	51.1	57.3	54.0
9	Base+YT	63.5	72.4	67.6	61.7	70.0	65.5	54.8	60.6	57.6	52.4	58.9	55.4
10	Base+YT+YM	63.9	72.6	68.0	62.1	70.4	66.0	55.2	61.0	57.9	52.8	59.1	55.8
11	Base+YT+YM+WP	66.1	75.4	70.4	62.9	72.4	67.3	57.7	64.4	**60.8**	55.1	61.6	**58.2**
12	Base+YT+YM+WP+FN	66.3	75.1	70.4	63.1	72.3	67.4	57.3	64.1	60.5	54.7	61.2	57.8
13	Base+YT+YM+WP+FN+VP	66.6	75.9	**70.9**	63.5	72.9	**67.9**	57.7	64.4	**60.8**	55.1	61.6	**58.2**
14	Base+YT+YM+WP+FN+VP+AP	66.4	75.7	70.7	63.3	72.9	67.8	57.6	64.3	**60.8**	55.0	61.5	58.1

Table 2: Results obtained by adding different types of features incrementally to the Baseline system.

and 51.5 (CEAF) on OntoNotes, and the CR model achieves F-measure scores of 66.1 (B^3) and 63.8 (CEAF) on ACE and 56.2 (B^3) and 54.0 (CEAF) on OntoNotes. Also, the results show that the CR model is stronger than the MP model, corroborating previous empirical findings (Rahman and Ng, 2009).

4.2.2 Incorporating World Knowledge

Next, we examine the usefulness of world knowledge for coreference resolution. The remaining rows in Table 1 show the results obtained when different types of features encoding world knowledge are applied to the Baseline system in isolation. The best result for each combination of annotation scheme, evaluation measure, and model is boldfaced.

Two points deserve mention. First, each type of features improves the Baseline, regardless of the coreference model, the evaluation measure, and the annotation scheme used. This suggests that all these feature types are indeed useful for coreference resolution. It is worth noting that in all but a few cases involving the FrameNet-based and Appositive-based features, the rise in F-measure is accompanied by a simultaneous rise in recall and precision. This is perhaps not surprising: as the use of world knowledge helps discover coreference links, recall increases;

1.	The Bush White House is breeding non-duck ducks the same way the **Nixon** White House did: It hops on an issue that is unopposable – cleaner air, better treatment of the disabled, better child care. **The President** came up with a good bill, but now may end up signing the awful bureaucratic creature hatched on Capitol Hill.
2.	**The tumor**, he suggested, developed when the second, normal copy also was damaged. He believed **colon cancer** might also arise from multiple "hits" on cancer suppressor genes, as it often seems to develop in stages.

Table 3: Example errors introduced by YAGO and FrameNet.

and as more (relevant) knowledge is available to make coreference decisions, precision increases.

Second, the feature types that yield the best improvement over the Baseline are YAGO TYPE and Noun Pairs. When the MP model is used, the best coreference system improves the Baseline by 1–1.3% (B^3) and 1.3–2.8% (CEAF) in F-measure. On the other hand, when the CR model is used, the best system improves the Baseline by 2.3–2.6% (B^3) and 1.7–2.2% (CEAF) in F-measure.

Table 2 shows the results obtained when the different types of features are added to the Baseline one after the other in the following order: YAGO TYPE, YAGO MEANS, Noun Pairs, FrameNet, Verb Pairs, and Appositives. In comparison to the results in Table 1, we can see that better results are obtained when the feature types are applied to the Baseline in combination than in isolation, regardless of the coreference model, the evaluation measure, and the annotation scheme used. The best-performing system, which employs all but the Appositive feature, outperforms the Baseline by 3.1–3.3% in F-measure when the MP model is used and by 4.1–4.8% in F-measure when the CR model is used. In both cases, the gains in F-measure are accompanied by a simultaneous rise in recall and precision.

In sum, our results suggest that (1) world knowledge can still offer benefits when used in combination with our Baseline feature set, which is richer than the one employed by Soon et al.; (2) the (more sophisticated) CR model makes more effective use of the available knowledge than the MP model; and (3) the different feature types provide complementary information for the two coreference models.

4.3 Example Errors

While the different types of features we considered improve the performance of the Baseline primarily via the establishment of coreference links, some of these links are spurious. In sentences 1 and 2 of Table 3, we show the spurious coreference links introduced by the CR model when YAGO and FrameNet are used, respectively. In sentence 1, while *The President* and *Bush* are coreferent, YAGO caused the CR model to establish the spurious link between *The President* and *Nixon* owing to the proximity of the two NPs and the presence of this NP pair in the YAGO TYPE relation. In sentence 2, FrameNet caused the CR model to wrongly posit *The tumor* and *colon cancer* as coreferent because these two NPs are the ARG0 arguments of *develop* and *arise*, which appear in the same frame in FrameNet.

5 Conclusions

We have examined the utility of three major sources of world knowledge for coreference resolution, namely, large-scale knowledge bases (YAGO, FrameNet), coreference-annotated data (Noun Pairs, Verb Pairs), and unannotated data (Appositives), by applying them to two learning-based coreference models, the mention-pair model and the cluster-ranking model, and evaluating them on documents annotated with the ACE and OntoNotes annotation schemes. When applying the different types of features in isolation to a Baseline system that does not employ world knowledge, we found that all of them improved the Baseline regardless of the underlying coreference model, the evaluation measure, and the annotation scheme, with YAGO TYPE and Noun Pairs yielding the largest performance gains. Nevertheless, the best results were obtained when they were applied in combination to the Baseline system. We conclude from these results that the different feature types we considered provide complementary world knowledge for the coreference resolvers, and while each of them offers fairly small gains, their cumulative benefits can be substantial.

Acknowledgments

We thank the three reviewers for their invaluable comments on an earlier draft of the paper. This work was supported in part by NSF Grant IIS-0812261.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the Linguistic Coreference Workshop at The First International Conference on Language Resources and Evaluation*, pages 563–566.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90.

David Bean and Ellen Riloff. 2004. Unsupervised learning of contextual role knowledge for coreference resolution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 297–304.

Eric Bengtson and Dan Roth. 2008. Understanding the values of features for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 294–303.

Volha Bryl, Claudio Giuliano, Luciano Serafini, and Kateryna Tymoshenko. 2010. Using background knowledge to support coreference resolution. In *Proceedings of the 19th European Conference on Artificial Intelligence*, pages 759–764.

Eugene Charniak. 1972. *Towards a Model of Children's Story Comphrension*. AI-TR 266, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Aron Culotta, Michael Wick, and Andrew McCallum. 2007. 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*, pages 81–88.

Hal Daumé III and Daniel Marcu. 2005. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 97–104.

Pascal Denis and Jason Baldridge. 2008. Specialized models and ranking for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 660–669.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 363–370.

Michael Fleischman, Eduard Hovy, and Abdessamad Echihabi. 2003. Offline strategies for online question answering: Answering questions before they are asked. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 1–7.

Aria Haghighi and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1152–1161.

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

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142.

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the Bell tree. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 135–142.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32.

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

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111.

Vincent Ng. 2007. Shallow semantics for coreference resolution. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pages 1689–1694.

Simone Paolo Ponzetto and Massimo Poesio. 2009. State-of-the-art NLP approaches to coreference resolution: Theory and practical recipes. In *Tutorial Abstracts of ACL-IJCNLP 2009*, page 6.

Simone Paolo Ponzetto and Michael Strube. 2006. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proceedings* *of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 192–199.

Sameer S. Pradhan, Wayne H. Ward, Kadri Hacioglu, James H. Martin, and Dan Jurafsky. 2004. Shallow semantic parsing using support vector machines. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 233–240.

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

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

Veselin Stoyanov, Nathan Gilbert, Claire Cardie, and Ellen Riloff. 2009. Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 656–664.

Veselin Stoyanov, Claire Cardie, Nathan Gilbert, Ellen Riloff, David Buttler, and David Hysom. 2010. Coreference resolution with Reconcile. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 156–161.

Fabian Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A core of semantic knowledge unifying wordnet and wikipedia. In *Proceedings of the World Wide Web Conference*, pages 697–706.

Olga Uryupina, Massimo Poesio, Claudio Giuliano, and Kateryna Tymoshenko. 2011. Disambiguation and filtering methods in using web knowledge for coreference resolution. In *Proceedings of the 24th International Florida Artificial Intelligence Research Society Conference*.

Xiaofeng Yang, Guodong Zhou, Jian Su, and Chew Lim Tan. 2003. Coreference resolution using competitive learning approach. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 176–183.

Xiaofeng Yang, Jian Su, Jun Lang, Chew Lim Tan, and Sheng Li. 2008. An entity-mention model for coreference resolution with inductive logic programming. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 843–851.