Coreference Resolution with ILP-based Weighted Abduction

Naoya Inoue[1] *Ekaterina Ovchinnikova*[2]
Kentaro Inui[1] *Jerry Hobbs*[2]
(1) Tohoku University, 6-6-05 Aoba, Aramaki, Aoba-ku, Sendai, 980-8579, Japan
(2) USC/ISI, 4676 Admiralty Way, Marina del Rey, CA 90292, USA
naoya-i@ecei.tohoku.ac.jp,katya@isi.edu,inui@ecei.tohoku.ac.jp,hobbs@isi.edu

ABSTRACT
This paper presents the first systematic study of the coreference resolution problem in a general inference-based discourse processing framework. Employing the mode of inference called weighted abduction, we propose a novel solution to the overmerging problem inherent to inference-based frameworks. The overmerging problem consists in erroneously assuming distinct entities to be identical. In discourse processing, overmerging causes establishing wrong coreference links. In order to approach this problem, we extend Hobbs et al. (1993)'s weighted abduction by introducing weighted unification and show how to learn the unification weights by applying machine learning techniques. For making large-scale processing and parameter learning in an abductive logic framework feasible, we employ a new efficient implementation of weighted abduction based on Integer Linear Programming. We then propose several linguistically motivated features for blocking incorrect unifications and employ different large-scale world knowledge resources for establishing unification via inference. We provide a large-scale evaluation on the CoNLL-2011 shared task dataset, showing that all features and almost all knowledge components improve the performance of our system.

KEYWORDS: weighted abduction, coreference resolution, Integer Linear Programming.

1 Introduction

In this paper, we explore coreference resolution in a discourse processing framework based on a mode of inference called *weighted abduction* (Hobbs et al., 1993). This framework is appealing because it is a realization of the observation that we understand new material by linking it with what we already know. It instantiates in natural language understanding the more general principle that we understand our environment by coming up with the best explanation for the observables in the environment. Hobbs et al. (1993) show that the lowest-cost abductive proof provides the solution to a whole range of natural language pragmatics problems, such as word sense disambiguation, anaphora and metonymy resolution, interpretation of noun compounds and prepositional phrases and detection of discourse relations. For examples of the application of weighted abduction to discourse processing see (Charniak and Goldman, 1991; Inoue and Inui, 2011; Ovchinnikova et al., 2011; Ovchinnikova, 2012).

If weighted abduction is applied to discourse processing, coreference links naturally follow as a by-product of constructing best explanations. In weighted abduction, coreference resolution is equal to unification of predications; see Sec. 3.1. Similarly, if deductive model building is applied to discourse interpretation, coreference links result from the model minimality. Both inference approaches are based on the idea that predications having the same predicates describe the same situation and therefore their arguments can be assumed to be equal if no logical contradictions follow. If the necessary knowledge is missing from the knowledge base, both the deductive and the abductive procedures are likely to miss relevant coreference links and establish wrong links (overmerge entities). The *overmerging* problem is a serious obstacle in applying reasoning to discourse processing, because it leads to a large number of incorrect inferences; see (Ovchinnikova, 2012) for examples. There have been attempts to employ semantic similarity for merging predications in a deductive framework (Dellert, 2011) and attempts to use linguistically motivated constraints in order to prohibit incorrect unification in an abductive framework (Ovchinnikova et al., 2011; Ovchinnikova, 2012). However, the issue of overmerging was never systematically studied and the proposed solutions were never evaluated. In this paper, we investigate whether adding linguistically motivated features can help to block incorrect links in an inference-based framework.

A lot of effort in NLP was put into coreference resolution systems ranging from rule-based (Lee et al., 2011, etc.) to machine learning-based resolvers (Soon et al., 2001; Ng and Cardie, 2002; Fernandes et al., 2012, etc.); see (Ng, 2010) for a detailed survey. Coreference resolution may require deep understanding of text, access to world knowledge, and inference ability. For example, (Levesque, 2011) considers twin sentences such as *Ed shouted at Tim because he crashed the car* and *Ed shouted at Tim because he was angry*. In order to resolve coreference in these sentences one requires world knowledge about people shouting when being angry and people shouting at someone who made a mistake, e.g., crashed a car. Surprisingly, most of the contemporary coreference resolution systems including the winners of the CoNLL-2011 and CoNLL-2012 shared tasks (Lee et al., 2011; Fernandes et al., 2012) do not exploit any world knowledge. There exist attempts to resolve coreference based on world knowledge resources such as WordNet hierarchy, Wikipedia, semantic similarity, narravite chains (Ponzetto and Strube, 2006; Ng, 2007; Irwin et al., 2011; Rahman and Ng, 2012). Unfortunately, the corresponding resolvers were either not evaluated in large-scale challenges or did not show convincing performance in the challenges. Thus, the question remains open whether employing world knowledge can improve coreference resolution in large unfiltered corpora. In this paper, we investigate whether adding world knowledge for establishing more coreference links can

improve coreference resolution. In the world knowledge employed, our work is most similar to the study on twin sentences presented in (Rahman and Ng, 2012). However, instead of using world knowledge for generating features in a machine learning framework, we explore inference-based discourse processing. Regarding inference, our method may seem related to the coreference resolution research based on Markov Logic Networks (MLNs) (Poon and Domingos, 2008; Song et al., 2012). However, previous MLN-based work on coreference resolution does not incorporate inference rules based on world knowledge.

The key contributions of our work are the following. First, we propose a novel solution to the overmerging problem in an inference-based framework. We extend (Hobbs et al., 1993)'s weighted abduction in order to accommodate unification weights and show how to learn the weights by applying machine learning techniques. For making large-scale processing and parameter learning in an abductive logic framework feasible, we employ a new efficient implementation of weighted abduction based on the Integer Linear Programming technique (Inoue and Inui, 2011).[1] Second, we propose several linguistically motivated features for blocking incorrect unifications and we employ different large-scale world knowledge resources for establishing unification via inference. Third, we report on a large-scale evaluation showing that all features and knowledge components improve the performance.

The structure of this paper is as follows. In Sec. 2, we introduce weighted abduction and its ILP-based implementation. Section 3 describes our discourse processing pipeline based on weighted abduction and discusses the overmerging problem, our solution to it, and types of knowledge we employ for generation of features and axioms. Section 4 presents the experiments on coreference resolution. The final section concludes the paper.

2 Abductive Inference

2.1 Weighted Abduction

Abduction is inference to the best explanation. Formally, logical abduction is defined as follows:

Given: Background knowledge B, observations O, where both B and O are sets of first-order logical formulas,

Find: A hypothesis H such that $H \cup B \models O, H \cup B \not\models \bot$, where H is a set of first-order logical formulas. We say that p is *hypothesized* if $H \models p$, and that p is *explained* if $(\exists q)\, q \rightarrow p \in B$ and q is hypothesized or explained.

Typically, there exist several hypotheses H explaining O. Each of them is called a *candidate hypothesis*. To rank candidate hypotheses according to plausibility, we use the framework of *weighted abduction* as defined by Hobbs et al. (1993). In this framework, observation O is a conjunction of propositions existentially quantified with the widest possible scope. Each proposition has a positive real-valued cost. We use the notation $P^{\$c}$ to indicate that proposition P has cost c and $cost(P)$ to represent the cost of P.

The background knowledge B is a set of first-order logic formulas of the form $P_1^{w_1} \wedge ... \wedge P_n^{w_n} \rightarrow Q_1 \wedge ... \wedge Q_m$. All variables occurring in the antecedent of such axioms are universally quantified with the widest possible scope. Other variables are existentially quantified within the scope of the universal quantifiers. Propositions in the antecedents are assigned positive real-valued *weights*. We use the notation P^w to indicate that proposition P has weight w.

[1] There has been work on applying ILP to coreference (Finkel and Manning, 2008; Denis and Baldridge, 2009), but with no relationship with logical inference.

The two main inference operations in weighted abduction are backward chaining and unification. *Backward chaining* is the introduction of new assumptions given an observation and background knowledge. For example, given $O = \exists x(q(x)^{\$10})$ and $B = \{\forall x(p(x)^{1.2} \rightarrow q(x))\}$, there are two candidate hypotheses: $H_1 = \exists x(q(x)^{\$10})$ and $H_2 = \exists x(p(x)^{\$12})$. In weighted abduction, a *cost function* f is used to calculate assumption costs. The function takes two arguments: costs of the propositions backchained on and weight of the assumption. Usually, a multiplication function is used, i.e. $f(c,w) = c \cdot w$, where c is the cost of the propositions backchained on and w is weight of the corresponding assumption. For example, if $q(x)$ costs \$10 and w of p is 1.2 in the example above, then assuming p in H_2 costs \$12.

Unification is the merging of propositions with the same predicate name by assuming that their arguments are same. For example, $O = \exists x, y(p(x)^{\$10} \wedge p(y)^{\$20} \wedge q(y)^{\$10})$. There is a candidate hypothesis $H = \exists x, y(p(x = y)^{\$10} \wedge x = y^{\$0} \wedge q(x = y)^{\$10})$, where $p(x)^{\$10}$ and $p(y)^{\$20}$ are merged by assuming $x = y$ (called *variable unification assumption*). Hobbs et al. (1993) assign the smallest cost to the result of the unification (i.e. \$10), and zero cost to the variable unification assumption. This principle often causes incompatible entities to be identified (e.g., a dog and a cat) on the basis of slender evidence, since unification always reduces the cost of hypothesis. In order to address this issue, we propose to assign a cost to the variable unification assumption. We use a weighted feature function to assign the cost, where the appropriate weights are learnable from the dataset (see Sec. 3 for further details).

Both operations (backchaining and unification) can be applied to an observation as many times as possible to generate a possibly infinite set of candidate hypotheses. Henceforth, we denote $\mathscr{H}_O$ to represent a set of all possible candidate hypotheses for O. Weighted abduction defines a *cost* of candidate hypothesis H as $cost(H) = \sum_{h \in H} cost(h)$, where h is an atomic conjunct in H also called an *elemental hypothesis* (e.g., $p(x)$ in the above H). In this framework, minimum-cost explanations are best explanations.

2.2 ILP-based Weighted Abduction

Recently, an implementation of weighted abduction based on Integer Linear Programming (ILP) was developed by Inoue and Inui (2011). In this approach the abductive reasoning problem is formulated as an ILP optimization problem. We adopt this solution since (i) the ILP-based reasoner is significantly more efficient than existing implementations of weighted abduction (Inoue and Inui, 2011), and (ii) its declarative nature makes it is highly extensible (Sec. 3).

Given B and O, the framework first enumerates set P of *potential elemental hypotheses* (atomic assumptions). Then it generates ILP variables and constraints based on this set to represent all possible *candidate hypotheses*. The four main ILP variables are $h_p \in \{0,1\}$, $r_p \in \{0,1\}$, $u_{p,q} \in \{0,1\}$, and $s_{x,y} \in \{0,1\}$, where p, q are potential elemental hypotheses and x, y are first-order logical variables or constants used in P. h_p is used to represent whether p is hypothesized ($h_p = 1$) or not ($h_p = 0$). r_p is used to represent whether p pays its cost ($r_p = 0$) or not (p is explained, $r_p = 1$). The ILP objective function is as follows.

$$\text{min. } cost(H) = \sum_{p \in \{p | p \in P, h_p = 1, r_p = 0\}} cost(p) \tag{1}$$

Thus, the cost of H is the sum of the costs of $p \in P$, such that p is included in the hypothesis ($h_p = 1$) and is *not* explained ($r_p = 0$). That is, the backchaining bottoms out in p.

The space of candidate hypotheses is restricted by several ILP constraints. For example, one of the constraints allows us to set $r_p = 1$ (p does not pay its cost) only if at least one proposition $q \in Q$, where Q is a set of propositions that explain p, is hypothesized ($h_q = 1$). The ILP formulation of this constraint is $r_p \leq \sum_{q \in Q} h_q$.

In order to represent unification of two propositions p and q we introduce variables u and s, such that $u_{p,q} = 1$ if p and q are unified and $u_{p,q} = 0$ otherwise; $s_{x,y} = 1$ if variables x and y are set to be equal and $s_{x,y} = 0$ otherwise. Additional constraints are defined on these variables. For example, $p(x_1, x_2, ..., x_n)$ and $p(y_1, y_2, ..., y_n)$ can be unified ($u_{p(x_1,x_2,...,x_n),p(y_1,y_2,...,y_n)} = 1$) only if their corresponding arguments are assumed to be equal (for all $i \in \{1, 2, ..., n\}$, $s_{x_i,y_i} = 1$). This is captured by the following ILP constraint: $n \cdot u_{p(x_1,x_2,...,x_n),p(y_1,y_2,...,y_n)} \leq \sum_{i=1}^{n} s_{x_i,y_i}$.

Formulation of the ILP constraints corresponding to variable inequality is rather straightforward.[2] For each pair of variables x and y such that $x \neq y \in P$, the following equality is introduced: $s_{x,y} = 0$.

3 Coreference Resolution in ILP-based Abductive Framework

3.1 Abduction for Discourse Processing

Abductive reasoning can be used to recover implicit information from natural language texts. The implicit information includes semantic relations between discourse entities, anaphoric relations, character's intentions, etc; see (Hobbs et al., 1993) for detailed examples.

A logical form (LF) of a text represents observations, which need to be explained by background knowledge. In our discourse processing pipeline, a text is first input to the English parser *Boxer* (Bos, 2008). For each segment, the parse produced by *Boxer* is a first-order fragment of the DRS language used in Discourse Representation Theory (Kamp and Reyle, 1993). An add-on to *Boxer* converts the DRS into a logical form in the style of (Hobbs, 1985).

The LF is a conjunction of propositions, which have generalized eventuality arguments that can be used for showing relationships among the propositions. According to (Hobbs, 1985), any predication in the logical notation has an extra argument, which refers to the "condition" of that predication being true. Thus, in the logical form $John(e_1, j) \wedge run(e_2, j)$ for the sentence *John runs*, e_2 is a running event by John and e_1 is a condition of j being named "John".

In the context of discourse processing, we call a hypothesis explaining a logical form *an interpretation* of this LF. The interpretation of the text is carried out by an abductive system. The system tries to prove the logical form of the text, allowing assumptions where necessary. Where the system is able to prove parts of the LF, it is anchoring it in what is already known from the overall discourse or from a knowledge base. Where assumptions are necessary, it is gaining new information.

Let us illustrate the procedure with an example implying coreference resolution. Suppose we need to interpret the text *John gave Bill a book; he was happy to have it*. A simplified logical form of this sentence is as follows:

$John(e_1, x_1) \wedge give(e_2, x_1, x_2, x_3) \wedge Bill(e_3, x_2) \wedge book(e_4, x_3) \wedge he(e_5, x_4) \wedge have(e_6, x_4, x_5) \wedge it(e_7, x_5)$

Suppose our knowledge base contains the following axioms

[2] See (Inoue and Inui, 2012) for the ILP representation of negated propositions.

(1) $give(e_1, x_1, x_2, x_3) \rightarrow get(e_2, x_2, x_3)$
(2) $get(e_1, x_1, x_2) \rightarrow have(e_2, x_1, x_2)$

Given these axioms, we can backchain on $have(e_6, x_4, x_5)$ to $give(e_0, u, x_5, x_4)$. After unifying this proposition with $give(e_2, x_1, x_2, x_3)$, we can infer the equality $x_2 = x_4$ and $x_3 = x_5$, which corresponds to linking *he* to *Bill* and *it* to *book* in the sentence.

3.2 Weighted Unification

Frequently, the lowest-cost interpretation results from identifying two entities with each other, so that their common properties only need to be proved or assumed once. This feature of the algorithm is called "unification", and is one of the principal methods by which coreference is resolved. A naive approach to coreference in an inference-based framework is to unify propositions having the same predicate names unless it implies logical contradictions (Hobbs et al., 1993; Bos, 2011). However, in situations when knowledge necessary for establishing contradictions is missing, the naive procedure results in *overmerging*. For example, given $O = animal(e_1, x) \wedge animal(e_2, y)$, weighted abduction incorrectly assumes x equals y even when $dog(e_3, x)$ and $cat(e_4, y)$ are observed. For *John runs and Bill runs*, with the observations $O = John(e_1, x) \wedge run(e_2, x) \wedge Bill(e_3, y) \wedge run(e_4, y)$, weighted abduction assumes John and Bill are the same individual just because they are both running. If we had complete knowledge about disjointness (*dog* and *cat* are disjoint, people have unique first names), the overmerging problem might not occur because of logical contradictions. However, it is not plausible to assume that we would have an exhaustive knowledge base.

In this study, we impose costs on variable unification assumptions in order to avoid the overmerging. If the unification weights are introduced, unification does not always reduce the overall cost of the hypothesis anymore, which loosens the assumption that all propositions with the same predicate names are coreferential.

How can we define unification weights? The cost of a variable unification assumption, say $x = y$, depends on the properties of x and y. For example, it depends on lexico-syntactic properties. If *x is different from y* or *x writes y* are observed in a text then its unlikely that x and y refer to the same entity. At the same time, observations like $dog(x) \wedge animal(y)$ serve as an evidence that x and y might be coreferential.

We extend the ILP-based weighted abduction framework developed by (Inoue and Inui, 2011) and use a feature-based linear function $\phi(x, y)$ to determine a cost of $x = y$. Features are based on various types of knowledge (see Sec. 3.3). In order to compute the weight of each feature, we parametrize the feature function by a n-dimensional real-valued weight vector $\mathbf{w} = \{w_1, w_2, ..., w_n\}$, and propose to tune $\mathbf{w}$ in a supervised manner.

In order to exploit the cost of variable unification assumptions in the reasoning process, we extend the ILP-based objective function for weighted abduction equation (1) as follows:

$$\text{min. } cost(H; \mathbf{w}) = \frac{1}{Z} \sum_{p \in \{p | p \in P, h_p = 1, r_p = 0\}} cost(p) + \frac{1}{|T_P|^2} \sum_{x,y \in T_P} \sum_{i}^{n} w_i \cdot f_{x,y,i}, \qquad (2)$$

where Z is a total cost of observations, and T_P is a set of logical atomic terms that appear in the set of potential elemental hypotheses, and $f_{x,y,i} \in \{0, 1\}$ is a newly introduced ILP variable that denotes the value of i-th feature for x, y. We normalize each term so that the size of observations and the number of logical terms does not affect to the strength of each term.

The features can be designed by a user. The value of each feature depends on the presence of certain propositions in the corresponding candidate hypothesis. For example, the value of feature $f_{x,y,i}$ can depend on the presence of $dog(x)$ and $cat(y)$. This dependence is represented by the following ILP constraint: $-m+1 \leq m \cdot f_{x,y,i} - \sum_{j=1}^{m} h_{p_m} \leq 0$, where h is an ILP variable (see Sec. 2.2) and $p_1, p_2, ..., p_m$ are propositions on which $f_{x,y,i}$ depends (i.e. $f_{x,y,i} = 1 \Leftrightarrow h_{p_1} = 1 \wedge h_{p_2} = 1 \wedge ... \wedge h_{p_m} = 1$).

Weight Learning

In order to train weight vector $\mathbf{w}$, we employ the modified version of the Passive-Aggressive (PA) algorithm (Crammer et al., 2006), which is a supervised large-margin online learning algorithm applicable to a wide range of linear classifiers ranging from binary classifiers to structured predictors. The original PA algorithm requires the complete set of gold standard labels to be present in the training set. In our case, however, the training set is annotated just with the coreference links (unification sets), but not with other assumptions supporting unification. For example, $dog(x) \wedge animal(y)$ will be annotated with $x = y$, but not with $dog(y)$. Therefore we modified the original algorithm for learning weights from a partial gold standard.

Algorithm 1 depicts our learning algorithm. Every time we receive a training instance (O, H_t) from a set $\mathbb{D}$ of training instances, where O is an observation and H_t is a set of gold standard variable unification assumptions for O, we first find the lowest-cost hypothesis $\hat{H}$ given the current weight vector (line 3). If variable unification assumptions made in $\hat{H}$ are inconsistent with H_t (e.g., *dog* and *animal* are unified in $\hat{H}$, but not in H_t), we train the weight vector (line 5–7). In order to train the vector, we find the lowest-cost hypothesis $\overline{H}$ among candidate hypotheses that *are consistent with* H_t (line 5). To get $\overline{H}$, we add ILP constraints for all $x = y$ in $H_t(s_{x,y} = 1)$ and for all $x \neq y$ in $H_t(s_{x,y} = 0)$ to the ILP optimization problem.

The new weight vector $\mathbf{w}$ should satisfy the following conditions: (i) $cost(\overline{H}; \mathbf{w})$ is less than $cost(\hat{H}; \mathbf{w})$ by at least a margin $\Delta(\hat{H}, \overline{H})$, and (ii) the difference between current weight vector $\mathbf{w}'$ and new weight vector $\mathbf{w}$ is minimal. In line 6, we calculate how much $\mathbf{w}$ should be corrected, where C is a parameter of the PA algorithm that is the aggressiveness of weight updates. $\phi(\hat{H})$ and $\phi(\overline{H})$ are the sums of feature vectors for variable unification assumptions in $\hat{H}$ and $\overline{H}$ respectively. $\Delta(\hat{H}, \overline{H})$ is a loss function that measures how different $\hat{H}$ and $\overline{H}$ are. The more different $\hat{H}$ and $\overline{H}$ are the larger an ensured margin is. In our experiments, we use the loss function $\Delta_P(\hat{H}, \overline{H}) = W_O / T_O$, where T_O is the total number of pairs of logical atomic terms in the observation and W_O is the total number of variable unification assumptions for observed logical terms in $\hat{H}$ that disagrees with $\overline{H}$. We implemented this training algorithm in a distributed learning framework (McDonald et al., 2010).

3.3 Features

Each feature we use is defined for pairs of unifiable variables (v_1, v_2). The features are summarized in Table 1.

Incompatible properties If two entities have incompatible properties, they are unlikely to be identical. We use WordNet antonymy (*black – white*) and sibling relation (*cat – dog*) to derive incompatible properties. Moreover, we assume that two proper names not belonging to the same WordNet synset are unlikely to refer to the same entity. Correspondingly, we generate three binary features *A*, *S*, and *P* (see Table 1).

Algorithm 1 Passive-Aggressive algorithm for partial gold standard dataset.

for all $i \in \{1, 2, ..., N\}$ **do**
  for all $(O, H_t) \in \mathbb{D}$ **do**
    $\hat{H} \leftarrow \arg\min_{H \in \mathcal{H}_O} cost(H; \mathbf{w})$
    if $\hat{H} \not\models H_t$ **then**
      $\overline{H} \leftarrow \arg\min_{H \in \mathcal{H}_O} cost(H; \mathbf{w})$ s.t. $H \models H_t$
      $\tau \leftarrow \min(C, \frac{cost(\overline{H};\mathbf{w}) - cost(\hat{H};\mathbf{w}) + \Delta(\overline{H},\hat{H})}{||\phi(\hat{H}) - \phi(\overline{H})||^2})$
      $\mathbf{w} \leftarrow \mathbf{w} + \tau(\phi(\overline{H}) - \phi(\hat{H}))$
    end if
  end for
end for

Feature type	Feature
Incompatible properties	$A(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p_1(.., v_1, ..), p_2(.., v_2, ..): p_1, p_2 \text{ are WN antonyms;} \\ 0 \text{ otherwise} \end{cases}$ $S(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p_1(.., v_1, ..), p_2(.., v_2, ..): p_1, p_2 \text{ are WN siblings;} \\ 0 \text{ otherwise} \end{cases}$ $P(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p_1(e_1, v_1), p_2(e_2, v_2): p_1, p_2 \text{ are proper names,} \\ \text{not in the same WN synset;} \\ 0 \text{ otherwise} \end{cases}$
Conditional unification	$CU(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p_1(v_1, x_1, .., x_n), p_2(v_2, y_1, .., y_n): \\ p_1, p_2 \text{ are frequent predicates} \\ \text{and } \forall i \in \{1, .., n\} : s_{x_i, y_i} = 1; \\ 0 \text{ otherwise} \end{cases}$
Argument inequality	$SA(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p(.., v_1, .., v_2, ..); \\ 0 \text{ otherwise} \end{cases}$ $EA(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p(v_1, .., e_1, ..), p(v_2, .., e_2, ..): s_{v_1,v_2} \wedge s_{e_1,e_2} = 0; \\ 0 \text{ otherwise} \end{cases}$
Explicit non-identity	$NI(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p(e, v_1, v_2): p \text{ is a non-identity predicate;} \\ 0 \text{ otherwise} \end{cases}$
Functional relations	$FR(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p(e_1, v_1, x_1), p(e_2, v_2, x_2): \\ p \text{ is a functional relation predicate} \\ \text{and } x_1 \neq x_2 \text{ and } v_1 = v_2; \\ 0 \text{ otherwise} \end{cases}$
Modality	$M(v_1, v_2) = \begin{cases} 1 \text{ if } \lvert MCPred(v_1) \cap MCPred(v_2) \rvert = \emptyset; \\ 0 \text{ otherwise} \end{cases}$
Common properties	$CP_1(v_1, v_2) = \lvert CPred(v_1, v_2) \rvert$, $CP_2(v_1, v_2) = \sum_{p \in CPred(v_1, v_2)} Freq(p)$ $CP_3(v_1, v_2) = \sum_{p \in CPred(v_1, v_2)} WNAbst(p)$
Derivational relation	$DR(v_1, v_2) = \begin{cases} 1 \text{ if } \exists p_1(v_1, ..), p_2(v_2, ..): \\ p_1, p_2 \text{ are derivationally related;} \\ 0 \text{ otherwise} \end{cases}$

Table 1: Summary of the feature set.

Conditional unification If two entities have very frequent common properties, these properties usually do not represent a good evidence for the entities to be identical. For example, given *John goes* and *he goes*, it might be incorrect to assume that *John* and *he* are coreferential just

because they are both going. We want to allow unification of frequent predications (e.g., *go*) only if there is other evidence for their arguments to be unified. In order to capture this idea, we introduce binary feature CU and compute its value as follows: If v_1 and v_2 occur as first arguments of propositions $p_1(v_1, x_1, .., x_n), p_2(v_2, y_1, .., y_n)$, such that p_1, p_2 are frequent predicates, and $\forall i \in \{1, .., n\} : s_{x_i, y_i} = 1$ (where s is an ILP variable, see Sec. 2.2) then $CU(v_1, v_2) = 1$; otherwise $CU(v_1, v_2) = 0$.

Argument inequality We use two argument constraints to generate features. First, we assume that arguments of the same proposition usually cannot refer to the same entity. Reflexive verbs represent an exception (e.g., *John cut himself*), but we assume that these cases are resolved by the *Boxer* semantic parser (see Sec. 3.1) and do not require inference. We create binary feature SA and compute its value as follows: If v_1 and v_2 occur as arguments of the same proposition then $SA(v_1, v_2) = 1$; otherwise $SA(v_1, v_2) = 0$.

One more feature we introduce concerns event variables. For example, given the sentences *John said that Mary was reading* and *John said that he was tired* we do not want to unify both *say* propositions, because in each case something different has been said. Predicates like *say* usually have clauses as their arguments. Unifying clauses just because they are arguments of the same predicate is often incorrect. In our framework, a clause is represented by an event variable, i.e. a variable which is a first argument of the head of the clause. We make the following assumption: If two unifiable propositions $p(v_1, .., e_1, ..), p(v_2, .., e_2, ..)$ have event variables as their arguments, then they are unlikely to be unified if the event arguments have not been already unified. We create binary feature EA and compute its value as follows: if (i) there are two unifiable propositions $p(v_1, .., e_1, ..), p(v_2, .., e_2, ..)$ that have event variables e_1, e_2 as non-first arguments, (ii) $e_1 \neq e_2$, and (iii) $v_1 = v_2$, then $EA(v_1, v_2) = 1$; otherwise $EA(v_1, v_2) = 0$.

Explicit non-identity We manually collected a set of 33 predicates indicating explicit non-identity, e.g., *similar to*, *different from*. Presence of these predicates in a logical form indicates that their second and third arguments are unlikely to refer to the same entity. We create binary feature NI and compute its value as follows: If there is $p(e, v_1, v_2)$ and p is a predicate indicating explicit non-identity then $NI(v_1, v_2) = 1$; otherwise $NI(v_1, v_2) = 0$.

Functional relations A binary relation r is functional if $\forall x, y_1, y_2 : r(x, y_1) \wedge r(x, y_2) \rightarrow y_1 = y_2$. For example, a person can be a son of exactly one man. Lin et al. (2010) automatically learn functional relations from a corpus and assign a confidence score to each extracted relation. We use the set of functional relations generated by Lin et al. (2010) in order to generate feature FR. We extract 1,661 functional relations from the dataset. We create a binary feature FR and compute its value as follows: if (i) there are two predicates $p(e_1, v_1, x_1), p(e_2, v_2, x_2)$, where p indicates a functional relation, (ii) $x_1 \neq x_2$, and (iii) $v_1 = v_2$ then $FR(v_1, v_2) = 1$; otherwise $FR(v_1, v_2) = 0$.

Modality We assume that two predications having different modality are unlikely to refer to the same entity. For example, given *John runs* and *he does not/might run*, *John* and *he* are unlikely to be coreferential. Let $MPred(v)$ be a set of predicates that represent the modality of event v. In our experiments, we consider three modality-denoting predicates produced by the *Boxer* semantic parser (nec, pos, not), and verbal predicates (e.g., *think*) as modality-denoting predicates. We create binary feature M and compute its value as follows: if there are two unifiable verbal propositions $p(v_1, ...), p(v_2, ...)$ and $|MPred(v_1) \cap MPred(v_2)| = \emptyset$ then $M(v_1, v_2) = 1$; otherwise $M(v_1, v_2) = 0$.

Common properties We assume that the more properties two entities share the more likely it is that they are identical. For example, given *John was jogging, while Bill was sleeping. He jogs every day*, *John* and *he* are likely to be coreferential, because they are both arguments of *jog*. Let $CPred(v_1, v_2)$ be a set of pairs of predicates p_1, p_2, such that v_1, v_2 occur at the same argument positions of p_1 and p_2 while p_1 are p_2 equal or they occurs in the same WordNet synset. We generate three types of real-valued features: $CP_1(v_1, v_2) = |CPred(v_1, v_2)|$, $CP_2(v_1, v_2) = \sum_{p \in CPred(v_1, v_2)} Freq(p)$, and $CP_3(v_1, v_2) = \sum_{p \in CPred(v_1, v_2)} WNAbst(p)$, where $Freq(p)$ is a word-frequency of p from the Corpus of Contemporary American English[3], and $WNAbst(p)$ is a level of abstraction of p in the WordNet hierarchy (the number of steps to the root).

Derivational relations We use WordNet derivational relations between nouns and verbs in order to link nominalizations and verbs. For example, given *Sales of cars grew. The growth followed year-to-year increases*, *grew* and *growth* are coreferential. We generate binary feature *DR* to capture these links (see Table 1).

3.4 Knowledge for Inference

The abductive reasoning procedure is based on a knowledge base consisting of a set of axioms. In the experiment described in this paper we employed the following background knowledge.

WordNet The dataset we use for evaluation (see Sec. 4) is annotated with WordNet (Fellbaum, 1998) senses. Given this annotation, we mapped word senses to WordNet synsets. Given WordNet relations defined on synsets, we generate axioms of the following form:

Hyperonymy, instantiation: $synset_1(s_1, x) \rightarrow synset_2(s_2, x)$
Causation, entailment: $synset_1(s_1, e_1) \rightarrow synset_2(s_2, e_2)$
Meronymy, membership: $synset_1(s_1, x_1) \rightarrow synset_2(s_2, x_2) \wedge of(x_1, x_2)$

We extracted 22,815 axioms from WordNet.

FrameNet We generated axioms mapping predicates with their arguments into FrameNet (Ruppenhofer et al., 2010) frames and roles. For example, the following axiom maps the verb *give* to the GIVING frame.

$\text{GIVING}(e_1) \wedge \text{DONOR}(e_1, x_1) \wedge \text{RECIPIENT}(e_1, x_2) \wedge \text{THEME}(e_1, x_3) \rightarrow give(e_1, x_1, x_3) \wedge to(e_2, e_1, x_2)$

Weights of these axioms are based on frequencies of lexeme-frame mappings in the annotated corpora provided by the FrameNet project. Moreover, we used FrameNet frame relations to derive axioms. An example of an axiomatized relation is given below.

$\text{GIVING}(e_1) \wedge \text{DONOR}(e_1, x_1) \wedge \text{RECIPIENT}(e_1, x_2) \wedge \text{THEME}(e_1, x_3) \rightarrow$
$\text{GETTING}(e_2) \wedge \text{SOURCE}(e_2, x_1) \wedge \text{RECIPIENT}(e_1, x_2) \wedge \text{THEME}(e_1, x_3)$

In order to generate the FrameNet axioms, we used the previous work on axiomatizing FrameNet by Ovchinnikova (2012). We generated 12,060 axioms from the dataset. In addition, we used a resource assigning possible lexical fillers disambiguated into WordNet synsets to FrameNet roles (Bryl et al., 2012). For example, the role THEME of the GIVING frame is mapped to synsets `object#n#1` and *thing#n#1*. Given this information, the following axiom is generated.

$thing\#n\#1(s, x) \rightarrow \text{GIVING}(e_1) \wedge \text{THEME}(e_1, x)$

[3] http://www.wordfrequency.info/

Weights of these axioms are based on the scores provided by Bryl et al. (2012). We generated 24,571 axioms from the dataset.

Narrative chains Similar to (Rahman and Ng, 2012), we employ narrative chains learned by Chambers and Jurafsky (2009), which were shown to have impact on resolving complex coreference; see (Rahman and Ng, 2012) for details. Narrative chains are partially ordered sets of events centered around a common protagonist that are likely to happen in a sequence. Knowledge about such sequences can facilitate coreference resolution. For example, given *Max fell, because John pushed him* we know that *Max* and *him* are coreferential, because we know that an object of the pushing event can be a subject of the falling event. For example, we generate the following axioms.

$Script\#1(s, e_1, x_1, u) \rightarrow \text{arrest}(e_1, x_1, x_2, x_3) \wedge police(e_2, x_1)$
$Script\#1(s, e_1, x_1, u) \rightarrow \text{charge}(e_1, x_1, x_2, x_3) \wedge police(e_2, x_1)$

Weights of these axioms are based on the scores provided by (Chambers and Jurafsky, 2009). We extract 1,391,540 axioms from the dataset.

3.5 Disambiguation of Named Entities

In the experiment on coreference resolution, we extended *Boxer*'s output with the information inferred by the *AIDA* tool. The *AIDA* tool (Yosef et al., 2011) is a framework for entity detection and disambiguation. Given a natural language text, it maps mentions of ambiguous names onto canonical entities like people or places, registered in a knowledge base like DBpedia (Bizer et al., 2009) or YAGO (Suchanek et al., 2008). For example, mentions *A. Einstein* and *Einstein* will be both mapped to the YAGO node *Albert_Einstein*. An add-on to our pipeline assigns the same variables to each two named entities disambiguated by *AIDA* into the same YAGO node.

4 Evaluation

We evaluate coreference resolution in our weighted abduction framework using the CoNLL-2011 shared task dataset (Pradhan et al., 2011). The CoNLL-2011 dataset was based on the English portion of the OntoNotes 4.0 data (Hovy et al., 2006). OntoNotes is a corpus of large scale annotation of multiple levels of the shallow semantic structure in text. The OntoNotes coreference annotation captures general anaphoric coreference. Note that OntoNotes captures explicit coreference links only, while our procedure also discovers implicit semantic overlap.

The CoNLL-2011 shared task was to automatically identify mentions of entities and events in text and to link the corefering mentions together to form entity/event chains. In our experiment, we do not identify mentions, but only compute precision and recall of the inferred coreference links given the mentions identified in the gold standard annotation.

In the CoNLL-2011 shared task, four metrics were used for evaluating coreference performance: MUC, B^3, CEAF, and BLANC. The evaluation metrics are described in (Pradhan et al., 2011). Each of the metric tries to address the shortcomings of the earlier metrics. MUC is the oldest metric; it has been criticized for not penalizing overmerging (Recasens and Hovy, 2010). Since one of the goals of this study is to reduce overmerging in our inference-based framework, this metric does not seem to be representative for us. The B^3 and CEAF metrics were also considered to produce counterintuitive results (Luo, 2005; Recasens and Hovy, 2010). BLANC, as the most recent evaluation metric, overcomes the drawbacks of MUC, B^3, and CEAF. The definition formula of BLANC given in (Recasens and Hovy, 2010) is replicated in Table 2, where

rc, wc, rn, wn indicate the number of right coreference links, wrong coreference links, right non-coreference links, and wrong non-coreference links correspondingly.

Score	Coreference	Non-coreference	Metric
P	$P_c = \frac{rc}{rc+wc}$	$P_n = \frac{rn}{rn+wn}$	BLANC-P $= \frac{P_c+P_n}{2}$
R	$R_c = \frac{rc}{rc+wn}$	$R_n = \frac{rn}{rn+wc}$	BLANC-R $= \frac{R_c+R_n}{2}$
F	$F_c = \frac{2P_cR_c}{P_c+R_c}$	$F_n = \frac{2P_nR_n}{P_n+R_n}$	BLANC $= \frac{F_c+F_n}{2}$

Table 2: Definition formula for BLANC.

We rely on BLANC when drawing conclusions, but present values of the other three evaluation metrics as well.

4.1 Results and Discussions

We intend to evaluate whether the introduction of linguistically motivated features (Sec. 3.3) and world knowledge (Sec. 3.4) enables us to outperform the naive inference-based approach implying that predications with the same names refer to the same entities. In order to evaluate the impact of each feature and knowledge component separately, we run ablation tests.

Note that for 145 of 6,894 sentences in the test set, no logical forms were produces by the *Boxer* semantic parser. Moreover, in the run employing WordNet-based inference, inference results could not be produced for 101 of 303 test texts because of the computational complexity of reasoning. In order to keep the comparison fair, we use evaluate all features and knowledge components on the same set of 202 texts, for which inference results were produced in all runs.

Table 3 represents the results of the ablation tests. We test the features listed in Table 1 as well as axioms extracted from WordNet (WN), FrameNet (FN), narrative chains (NC) and knowledge provided by AIDA (AI). All features representing incompatible properties are tested together (IP in Table 3). Similarly, all argument inequality features (AI) and common property features (CP) are tested together.

The first row represents results for the run without employing any features and knowledge resources. In the second run, world knowledge is employed without linguistic features. These two runs correspond to the original weighted abduction approach to unification implying unification of all predications having the same predicate names. We see that adding knowledge results in lower values of BLANC. This happens because of the overmerging problem increased by additional coreference links inferred with the help of the employed knowledge resources.

Then we test linguistic features intended to block incorrect unification (IP, CU, AI, NI, FR, M) one by one. Each of the features improves the BLANC values; conditional unification CU has the most significant impact.[4] The common property feature (CP) and the derivational relations feature (M) introduce additional unifications. Therefore we test them together with the best combination of the unification blocking features (IP+CU+AI+NI+FR+M). Both features have a positive impact as compared to the run employing just the unification blocking features. Now we test each world knowledge component using the best combination of features

[4]It is interesting to note that MUC and B^3 evaluation metrics represent completely the opposite picture, which supports the criticism of these metrics for tolerating overmerging (Recasens and Hovy, 2010).

(IP+CU+AI+NI+FR+M+CP). Each knowledge component except for WordNet has a positive impact in terms of BLANC as compared to the run using the best combination of all features.

Features								Inference				MUC			B^3			CEAFE			BLANC		
IP	*CU*	*AI*	*NI*	*FR*	*M*	*CP*	*DR*	WN	FN	NC	AI	R	P	F	R	P	F	R	P	F	R	P	F
												73.7	69.6	71.6	75.5	39.9	52.2	30.7	36.1	33.2	53.0	51.7	39.1
								✓	✓	✓	✓	72.3	68.6	70.4	74.6	41.6	53.4	32.3	37.1	34.5	52.3	51.3	39.9
✓												71.2	68.8	70.0	73.2	41.8	53.2	32.8	35.9	34.3	53.5	51.9	41.0
	✓											33.4	58.2	42.5	42.5	76.2	54.6	59.6	28.6	38.7	55.7	60.9	56.6
		✓										70.1	68.4	69.3	72.3	41.8	53.0	33.1	35.3	34.2	53.0	51.6	41.0
			✓									70.4	68.5	69.4	72.5	41.6	52.9	32.3	34.8	33.5	52.8	51.5	40.5
				✓								70.4	68.5	69.5	72.7	41.7	53.0	32.5	35.0	33.7	52.9	51.6	40.7
					✓							70.3	68.4	69.3	72.6	41.9	53.1	32.8	35.4	34.1	53.3	51.7	41.0
✓	✓	✓	✓	✓	✓	✓						39.4	62.0	48.2	46.6	74.1	57.2	59.6	30.9	40.8	58.4	61.6	59.4
✓	✓	✓	✓	✓	✓		✓					36.6	61.2	45.8	44.6	75.8	56.1	59.8	29.6	39.6	57.5	61.4	58.6
✓	✓	✓	✓	✓	✓	✓	✓	✓				36.2	60.3	45.2	44.5	75.3	56.0	59.7	29.6	39.6	57.4	61.2	58.5
✓	✓	✓	✓	✓	✓	✓	✓		✓			40.7	63.1	49.5	48.5	73.2	58.4	58.6	30.8	40.4	59.5	61.0	60.1
✓	✓	✓	✓	✓	✓	✓	✓			✓		40.1	63.0	49.0	47.4	74.0	57.8	59.1	30.8	40.5	59.0	61.5	59.9
✓	✓	✓	✓	✓	✓	✓	✓				✓	42.5	64.3	51.1	49.1	72.8	58.7	58.6	31.5	41.0	59.7	61.5	60.4
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	42.9	64.4	51.5	50.5	73.3	59.9	59.4	32.8	42.3	59.9	60.9	60.3

Table 3: Ablation tests of features and world knowledge.

The results of the ablation tests show significant improvement over the naive approach (by more than 20% F-measure), but can we claim that we solved the overmerging problem? We perform one more experiment in order to get a deeper understanding of the performance of our discourse processing pipeline in coreference resolution.

The best performance in the CoNLL-2011 shared task was achieved by the *Stanford NLP* system (Lee et al., 2011) that is a rule-based resolver encoding traditional linguistic constraints on coreference. We replicate the results of *Stanford NLP* as applied to the CoNLL-2011 dataset; see the first row in Table 4. We use the output of *Stanford NLP* only for those texts, which could be processed by our discourse processing pipeline, therefore the recall/precision values for *Stanford NLP* in Table 4 are lower than the original results published in (Lee et al., 2011).

We aim at checking whether enriching the output of the state-of-the-art coreference resolver with additional links inferred by our system using all features and all world knowledge will improve the performance. The evaluation of the "merged" output is presented in the second row of Table 4 (SNLP+WA). Unfortunately, the precision of SNLP+WA is lower than that of SNLP alone. This happens because adding world knowledge results in new coreference links, while the overmerging problem is not completely solved. SNLP discovers 2277 out of 7557 correct coreference links and 40247 out of 41527 correct non-coreference links. In the merged output, there are more correct coreference links (3065), but less correct non-coreference links (36959). Note that *Stanford NLP* performs noun phrase coreference resolution only, while our system is not restricted to noun phrases and can also discover implicit coreference links.

System	MUC			B^3			CEAFE			BLANC		
	R	P	F	R	P	F	R	P	F	R	P	F
SNLP	42.8	74.4	54.3	50.4	85.2	63.4	66.3	32.6	43.7	63.5	76.2	66.7
SNLP+WA	52.0	70.1	59.7	57.3	72.7	64.1	60.5	37.2	46.1	64.8	64.7	64.7

Table 4: Performance of the *Stanford NLP* system (SNLP) compared to performance of our weighted abduction engine enriched with *Stanford NLP* (SNLP+WA) output.

The main cause of overmerging is related incompatible properties. We anticipated the incom-

patible properties to have a more significant impact on precision than they actually had in the ablation tests. But in the current study, we consider only those properties to be incompatible which are expressed syntactically in the same way, e.g., *Japanese goods* vs. *German goods*. However, the same property can be expressed by a wide variety of syntactic constructions, e.g., *goods from Germany*, *goods produced in Germany*, *Germany produced goods* etc. In order to discover deeper contradictions, we have to work on normalization of the representation of properties, e.g., use *origin*:*Germany*:x instead of *German*(e, x) and *from*$(e_1, x, y) \wedge$*Germany*(e_2, y). FrameNet attempts to achieve such a normalization by using standardized frame and role names. Unfortunately, the limited coverage of the FrameNet resource (Shen and Lapata, 2007; Cao et al., 2008) does not allow us to solve the problem on a large scale.

Analyzing the results, we also found overmergings not implying any explicit contradictions. For example, in the sentence *He sat near him*, both *he* propositions are unlikely be coreferential, but our framework fails to capture that. Such overmergings might be blocked by explicit modeling of discourse salience. In the future, we plan to use existing discourse salience models (e.g., (Lappin and Leass, 1994)) to create real-valued salience features for weighted unification.

One more issue concerns the quality of the obtained interpretations. Our learning framework assumes that we can obtain optimal solutions, but we also exploit suboptimal solutions by imposing a timeout in this experiment. However, it has been reported that exploiting suboptimal solutions sometimes hurts performance (Finley and Joachims, 2008). In the future, we will address this problem using an approximate learning framework (e.g., (Huang et al., 2012)).

Conclusion and perspectives

In this paper, we investigated the overmerging problem in a general inference-based discourse processing pipeline using the mode of inference called weighted abduction. In our framework, resolving coreference is a by-product of constructing best interpretations of text. Coreference links naturally result from unifications of predications during the inference process. The naive approach to unification involves unifying predications with the same predicate names.

This paper presents the first systematic study of the overmerging problem resulting from naive unification. We proposed several linguistically motivated features for blocking incorrect unifications as well as employed different large-scale world knowledge resources for establishing unification via inference. We extended ILP-based weighted abduction in order to accommodate unification weights and showed how to learn the weights in a supervised manner. All features and almost all knowledge components proved to improve the performance of our system tested on a large state-of-the-art test dataset.

We cannot claim that the problem of overmerging has been solved, because we still discover overmerging of explicit anaphora produced by our system as compared to a state-of-the-art rule-based coreference resolver. However, the proposed framework presents a significant improvement over the naive approach (by over 20% of F-measure). Moreover, it is highly extensible for including more features and knowledge sources.

Acknowledgments

We would like to thank Johan Bos for helping us to set up the *Boxer* system for our experiments. We also thank the *Stanford NLP* team and the *AIDA* team for their cooperation. This work was partially supported by Grant-in-Aid for JSPS Fellows (22-9719) and Grant-in-Aid for Scientific Research (23240018).

References

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009). Dbpedia - a crystallization point for the web of data. *J. Web Sem.*, 7(3):154–165.

Bos, J. (2008). Wide-Coverage Semantic Analysis with Boxer. In Bos, J. and Delmonte, R., editors, *Proceedings of STEP*, Research in Computational Semantics, pages 277–286. College Publications.

Bos, J. (2011). A survey of computational semantics: Representation, inference and knowledge in wide-coverage text understanding. *Language and Linguistics Compass*, 5(6):336–366.

Bryl, V., Tonelli, S., Giuliano, C., and Serafini, L. (2012). A novel framenet-based resource for the semantic web. In *SAC*, pages 360–365.

Cao, D. D., Croce, D., Pennacchiotti, M., and Basili, R. (2008). Combining word sense and usage for modeling frame semantics. In *Proc. of STEP 2008*, pages 85–101.

Chambers, N. and Jurafsky, D. (2009). Unsupervised Learning of Narrative Schemas and their Participants. In *Proceedings of ACL*, pages 602–610.

Charniak, E. and Goldman, R. P. (1991). A Probabilistic Model of Plan Recognition. In *Proceedings of AAAI*, pages 160–165.

Crammer, K., Dekel, O., Keshet, J., and S. Shalev-Shwartz, Y. S. (2006). Online Passive-Aggressive Algorithms. pages 551–585.

Dellert, J. (2011). Challenges of Model Generation for Natural Language Processing. Master's thesis, University of Tübingen.

Denis, P. and Baldridge, J. (2009). Global Joint Models for Coreference Resolution and Named Entity Classification. In *Procesamiento del Lenguaje Natural 42*, pages 87–96, Barcelona: SEPLN.

Fellbaum, C., editor (1998). *WordNet: an electronic lexical database*. MIT Press.

Fernandes, E., dos Santos, C., and Milidiú, R. (2012). Latent structure perceptron with feature induction for unrestricted coreference resolution. In *CoNLL Shared Task 2012*, pages 41–48.

Finkel, J. R. and Manning, C. D. (2008). Enforcing transitivity in coreference resolution. In *Proceedings of ACL*, pages 45–48.

Finley, T. and Joachims, T. (2008). Training structural svms when exact inference is intractable. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 304–311, New York, NY, USA. ACM.

Hobbs, J. R. (1985). Ontological promiscuity. In *Proceedings of ACL*, pages 61–69, Chicago, Illinois.

Hobbs, J. R., Stickel, M., Martin, P., and Edwards, D. (1993). Interpretation as abduction. *Artificial Intelligence*, 63:69–142.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). Ontonotes: The 90% solution. In *Proceedings of HLT-NAACL 2006*, pages 57–60.

Huang, L., Fayong, S., and Guo, Y. (2012). Structured perceptron with inexact search. In *HLT-NAACL*, pages 142–151.

Inoue, N. and Inui, K. (2011). ILP-Based Reasoning for Weighted Abduction. In *Proceedings of AAAI Workshop on Plan, Activity and Intent Recognition*.

Inoue, N. and Inui, K. (2012). Large-scale Cost-based Abduction in Full-fledged First-order Predicate Logic with Cutting Plane Inference.

Irwin, J., Komachi, M., and Matsumoto, Y. (2011). Narrative schema as world knowledge for coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 86–92, Portland, Oregon, USA. Association for Computational Linguistics.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Studies in Linguistics and Philosophy. Kluwer, Dordrecht.

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

Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., and Jurafsky, D. (2011). Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, CONLL Shared Task '11, pages 28–34, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Lin, T., Mausam, and Etzioni, O. (2010). Identifying Functional Relations in Web Text. In *EMNLP*, pages 1266–1276.

Luo, X. (2005). On coreference resolution performance metrics. In *HLT/EMNLP*, pages 25–32.

McDonald, R., Hall, K., and Mann, G. (2010). Distributed training strategies for the structured perceptron. In *NAACL2010*, pages 456–464.

Ng, V. (2007). Shallow semantics for coreference resolution. In *Proceedings of IJCAI*, pages 1689–1694.

Ng, V. (2010). Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the ACL*, pages 1396–1411.

Ng, V. and Cardie, C. (2002). Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 104–111, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ovchinnikova, E. (2012). *Integration of World Knowledge for Natural Language Understanding*. Atlantis Press, Springer.

Ovchinnikova, E., Montazeri, N., Alexandrov, T., Hobbs, J. R., McCord, M., and Mulkar-Mehta, R. (2011). Abductive Reasoning with a Large Knowledge Base for Discourse Processing. In *Proceedings of IWCS*, pages 225–234, Oxford, UK.

Ponzetto, S. P. and Strube, M. (2006). Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 192–199.

Poon, H. and Domingos, P. (2008). Joint unsupervised coreference resolution with markov logic. In *Proceedings of EMNLP*, pages 650–659.

Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M., Weischedel, R., and Xue, N. (2011). CoNLL-2011 shared task: modeling unrestricted coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, CONLL Shared Task '11, pages 1–27, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rahman, A. and Ng, V. (2012). Resolving Complex Cases of Definite Pronouns: The Winograd Schema Challenge. In *Proceedings of EMNLP-CoNLL*, pages 777–789.

Recasens, M. and Hovy, E. (2010). BLANC: Implementing the Rand Index for Coreference Evaluation. *Journal of Natural Language Engineering*.

Ruppenhofer, J., Ellsworth, M., Petruck, M., Johnson, C., and Scheffczyk, J. (2010). FrameNet II: Extended Theory and Practice. Technical report, Berkeley, USA.

Shen, D. and Lapata, M. (2007). Using Semantic Roles to Improve Question Answering. In *Proceeding of EMNLP-CoNLL*, pages 12–21.

Song, Y., Jiang, J., Zhao, W. X., Li, S., and Wang, H. (2012). Joint learning for coreference resolution with markov logic. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1245–1254, Jeju Island, Korea. ACL.

Soon, W. M., Ng, H. T., and Lim, D. C. Y. (2001). A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.*, 27(4):521–544.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2008). YAGO: A Large Ontology from Wikipedia and WordNet. *J. Web Sem.*, 6(3):203–217.

Yosef, M. A., Hoffart, J., Bordino, I., Spaniol, M., and Weikum, G. (2011). Aida: An online tool for accurate disambiguation of named entities in text and tables. In *Proc. of the 37th Intl. Conference on Very Large Databases (VLDB 2011)*, pages 1450–1453.