Improving a Coreference Resolution System using Distributional Semantics

Verbesserung eines Systems zur Koreferenzauflösung mittels Distributioneller Semantik
Bachelor-Thesis von Tim Feuerbach
September 2014

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Language
Technology

Improving a Coreference Resolution System using Distributional Semantics
Verbesserung eines Systems zur Koreferenzauflösung mittels Distributioneller Semantik

Vorgelegte Bachelor-Thesis von Tim Feuerbach

1. Gutachten: Prof. Dr. Chris Biemann
2. Gutachten: Martin Riedl

Tag der Einreichung:

Erklärung zur Bachelor-Thesis

Hiermit versichere ich, die vorliegende Bachelor-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. In der abgegebenen Thesis stimmen die schriftliche und elektronische Fassung überein.

Darmstadt, den 25. September 2014

(Tim Feuerbach)

Abstract

Coreference resolution is the task of clustering nominal phrases (mentions) in a document which refer to the same real-world entity. A common problem of automatic resolution systems is their inability to link bridging mentions, i.e. coreferent noun phrases with non-identical heads. Their detection requires semantic knowledge. Manually created databases like WordNet are only of limited use for the task, as they become outdated quickly and cover only a relatively small amount of words compared to the vocabulary of a web-sized corpus. In contrast, methods rooted in the field of Distributional Semantics allow to automatically mine semantic knowledge from arbitrary text. They build upon the hypothesis introduced by Zellig Harris that words in similar contexts bear similar meanings.

In this thesis, we present semantic machine learning features exploiting distributional knowledge for coreference resolution on English documents. The knowledge is taken from IS-A tuples and a distributional thesaurus. The latter is also used to model selectional restrictions. We integrate these features into the state-of-the-art Berkeley resolution system and show that they are able to improve its overall performance, measured using the average scores of the MUC, B^3 and CEAF_e evaluation metrics, significantly by 0.74 points on average. In particular, we increased the system's recall of bridging mentions by 10 percentage points. We discuss the decrease in precision caused by the new features, and find that a naïve approach to the removal of unwanted semantic relations in the thesaurus is insufficient to solve the problem.

Zusammenfassung

Koreferenzresolution bezeichnet das Clustering von Nominalphrasen (Referenten), welche sich auf dieselbe außersprachliche Entität beziehen. Ein häufiges Problem automatischer Resolutionssysteme stellt ihre Unzulänglichkeit dar, indirekte Anaphern (bridging mentions) aufzulösen, d. h. koreferente Nominalphrasen mit nicht wortidentischen Kernen. Ihre Erkennung erfordert semantisches Wissen. Manuell erstellte Datenbanken wie z. B. WordNet sind hierbei von begrenztem Nutzen, da sie schnell veralten, und verglichen mit dem Vokabular eines Korpus von Webgröße nur relativ wenige Wörter verzeichnen. Demgegenüber erlauben Methoden der Distributionellen Semantik, semantisches Wissen automatisch aus beliebigen Texten zu gewinnen. Sie gründen auf der von Zellig Harris formulierten These, dass Wörter in ähnlichen Kontexten eine ähnliche Bedeutung besitzen.

In dieser Thesis stellen wir semantische Features für einen maschinellen Lerner vor, welcher Koreferenzen in englischsprachigen Texten auflöst. Die Features greifen auf distributionelles Wissen zurück, welches IS-A-Tupeln und einem distributionellem Thesaurus entstammt. Letzterer wird zudem verwendet, um selektionale Restriktion zu modellieren. Wir integrieren diese Features in das State-of-the-Art-Resolutionssystem der Berkeley University und verbessern damit dessen Gesamtleistung, gemessen mithilfe der Evaluationsmetriken MUC, B^3 und CEAF_e, signifikant um durchschnittlich 0.74 Prozentpunkte. Insbesondere konnten wir den Anteil der korrekt aufgelösten indirekten Anaphern um zehn Prozentpunkte erhöhen. Wir gehen auf die Verringerung der Systempräzision durch die distributionellen Features ein, und stellen fest, dass eine simple Methode zur Entfernung ungewünschter semantischer Relationen aus dem Thesaurus das Problem nicht löst.

Contents

1 Introduction

When humans acquire information, they do so by reading texts, listening to recorded speech, or by other means of communication. While it seems to have done a good job during the previous millennia of our existence, the process appears to be slow compared to the speed at which computer systems can search millions of sentences for expressions. For example, to write an essay on the murders of noblemen in plays from the 19th century, one has to read them word by word to find those of interest. As a consequence, we would limit ourselves to a select few. It would be convenient if we could just hand over that question to an information retrieval system, which would yield the respective passages in return, but such a system does not exist yet. That is because the interpretation of utterances poses a difficult challenge for computers. To answer our example query, the system has to find all stage directions and dialogue speaking of murder, and to look up the unfortunate's profession in the *dramatis personæ*. However, the act could be hidden in an expression like *he died by her hand*, in which the person is not mentioned directly. The system must link the pronoun *he* to the right name to extract the information.

The latter requires *coreference resolution*, the task of identifying and grouping recurring mentions of the same entity. Consider the following example:

(1) Sarah is on vacation. Look at her pictures, she took these photos in Spain.

Here, *Sarah*, *her* and *she* refer to one entity, while *her pictures* and *these photos* refer to another.

Unlike in the early years of natural language processing, when coreference resolution was performed using hand-written rules based on linguistic theories modeling the discourse, modern systems most often take a machine learning approach (cf. Ng 2010). Given training data in which coreference is annotated, they learn the significance of features like exact string matching or sentence distance between two mentions. Durrett and Klein (2013) have recently shown that a machine learner with a feature set capturing only shallow lexical, syntactic and semantic information outperforms current state-of-the-art resolution systems. However, their system is unable to detect so called *bridging* links (Vieira and Teufel 1997) like *George Clooney – the actor*, which require world knowledge to be resolved. Durrett and Klein experimented with different knowledge resources, but noticed only a small gain in performance, and therefore declared the incorporation of semantics an "uphill battle". Even worse, the use of resources like WordNet (Miller 1995)[1] to compute the semantic similarity of candidates may actually decrease the performance, as observed by some contestants in the CoNLL-2011 shared task of modeling unrestricted coreference (see Section 5.1.1).

Being a manually crafted database, WordNet cannot contain every possible word in the English language and is therefore prone to out-of-vocabulary errors (Agirre et al. 2009). Moreover, being a general purpose lexicon (Hearst 1998), it contains only a few proper nouns and no domain-specific terms. Thus, to tackle bridging coreference, larger knowledge resources are required. They should preferably be compiled in an unsupervised or semi-supervised fashion, since the manual acquisition of data is costly and time-consuming.

Another phenomenon that requires semantic knowledge to be properly addressed is that of *selectional restrictions* (cf. Jurafsky and Martin 2014, p. 639). While resulting in grammatically correct sentences, some noun phrases tend not to be used as arguments of certain verbs. For example, the proper English sentence *the water applauded* makes no sense outside the world of fiction, since we expect something that is capable of applauding to be animate. This kind of information is useful when resolving pronouns, which by themselves only carry information about the referred entity's number and gender. For example, both state-of-the-art systems by the Stanford (Lee et al. 2013) and Berkeley University (Durrett and Klein 2013) incorrectly treat the mentions *the children*, *them* and *they* as the same entity in the following sentence:

(2) The children love lemonades. They drink them every day.

A useful resource for information on selectional restriction is FrameNet (Baker et al. 1998), in which verbs are organized in *frames*, concepts of events in which the verb's arguments are assigned roles according to the context. For example, the verb *to drink* is assigned to the 'Ingestion' frame, with the object being an 'Ingestible'. Unfortunately, only a few annotated example sentences are available per frame, so the fact that lemonades are ingestible has to be derived by other means. One could, of course, look up the verb's arguments in WordNet like Jurafsky and Martin (2014, p. 641) do exemplarily, but that leads us back to the sparse data problem mentioned above.

[1] A digital lexicon containing English words, distinguished by their sense and arranged in a hyponym taxonomy.

Distributional Semantics provides us with means to address both bridging coreference and selectional restrictions without the need of hand-crafted resources. The distributional hypothesis (Harris 1954) states that “words which are similar in meaning occur in similar contexts” (Rubenstein and Goodenough 1965). This correlation allows us to automatically extract semantic information from large, unannotated corpora. By mapping a word’s context to a set of features, the semantic similarity between two words can be calculated as a function of those features. For example, we expect *George Clooney* and *actor* to exhibit a higher semantic similarity than, say, *George Clooney* and *house*, which gives us a good clue which pair is more likely to corefer. On the other hand, if we record the co-occurrence frequency of each word and its context, we can enumerate the words that are prone to appear in a given context, and thus simulate selectional restriction. In this way, the (presumably) low number of times that *children* occurred in object position of *to drink* in our mined corpus can act as strong negative evidence of coreference.

In this thesis, we integrate features derived from distributional semantics into the state-of-the-art Berkeley coreference system (Durrett and Klein 2013).[2] For that we resort to *distributional thesauri* and pairs of words related by hyponymy extracted using IS-A-patterns, both made available by the JoBimText Project (Biemann and Riedl 2013b).[3] We show that these features can improve the overall performance by a small margin, and increase the number of resolved bridging anaphors remarkably. We identify the loss in precision as the main problem of using distributional methods in coreference resolution. Although features equal or similar to those we employ have all been evaluated in the task of coreference resolution (see Chapter 5), their impact on a state-of-the-art system is rarely studied in isolation.

The remainder of this thesis is organized as follows. Chapter 2 defines the task of coreference resolution formally and explains terms used in this work. We describe the Berkeley system in Chapter 3. Chapter 4 outlines the conception of distributional semantics, as well as its vector-space-, thesaurus- and pattern-based realizations, and describes the JoBimText framework we retrieved our semantic knowledge from. Chapter 5 gives an overview of previous work using hand-crafted and/or automatically acquired semantic information to resolve bridging anaphora and to model selectional restrictions. We introduce the training and testing data together with the setup used to acquire the distributional knowledge in Chapter 6. In Chapter 7, we explain our three groups of features in detail: lists of semantically similar words, IS-A hypernyms, and words likely to appear in a given context. Chapter 8 evaluates the performance of the extended model in comparison to the baseline in three aspects: coreference metrics, pairwise linkage and resolution of bridging mentions. Additionally, a manual analysis highlights and discusses the types of errors made by the system after the addition of our features. Finally, Chapter 9 concludes and proposes future work.

[2] `http://nlp.cs.berkeley.edu/projects/coref.shtml`

[3] `http://www.jobimtext.org`

2 Task and terminology

In this section, we introduce the vocabulary used throughout the rest of this work, and define the specific type of coreference we want to resolve (*identity coreference*). If not stated otherwise, the terms' definitions are taken from Jurafsky and Martin (2014, p. 710). We will refer to the following sentence for illustration purposes:[4]

(3) [Napoli] have backed [[their] Colombia defender Juan Zuniga] over [[his] challenge on [Neymar] that forced [the Brazil forward] out of [the World Cup] with [a fractured bone in [[his] back]]].

Given a document, we extract *mentions* by marking all noun phrases (NPs) that refer to a real-world entity. The set of mentions is a subset of, but not necessarily identical to the set of all noun phrases, since for example nouns that take part in idiomatic expressions like *it took the cake* are non-referring. In sentence (3), we marked the mentions with brackets, resulting in the following set: {*Napoli, their, their Colombia defender Juan Zuniga, his*$_1$*, his challenge* [. . .] *his back, Neymar, the Brazil forward, the World Cup, a fractured bone in his back, his*$_2$*, his back*} (subscripts are used to distinguish between identically worded phrases).

We adopt the formal definition of coreference from van Deemter and Kibble (2000):

Definition 1. Let Referent(m) denote the entity that is referred to by mention m. Then, m_1 and m_2 are said to corefer if and only if Referent(m_1) = Referent(m_2).

By this definition, coreference acts as an equivalence relation. The problem of coreference resolution can thus be formulated as partitioning the set of mentions into its true equivalence classes according to the coreference relation (Mitkov 2002, p. 5). The example's correct partition is as follows:

- {*Napoli, their*}
- {*their Colombia defender Juan Zuniga, his*$_1$}
- {*his challenge* [. . .] *his back*}
- {*Neymar, the Brazil forward, his*$_2$}
- {*the World Cup*}
- {*a fractured bone in his back*}
- {*his back*}

Unless a distinction is necessary, we will call these equivalence classes *entities* as well.

Mentions that refer to a mention appearing earlier in the document are called *anaphors*. We refer to entities containing only one mention and the mentions themselves as *singletons*. In example (3), *the World Cup*, *a fractured bone in his back* and *his back* are singletons.

Coreference should not be confused with anaphoricity, which is not an equivalence relation (van Deemter and Kibble 2000). While most anaphoric references are coreferential as well, this is not the general case, as we will see below.

Except in so called 'gold' settings, the set of mentions to partition is unknown to the system. Most often, a high recall approach is used by labeling all of a document's noun phrases as mentions, even though most of them are not coreferential.[5] Singletons are then removed from the final system output. This better reflects a real-world setting, since pre-labeled mentions already give away that they are referring, and an important part of the task involves the detection of spurious mentions.

An entity can be viewed as a *coreference chain* by arranging its mentions in the order they appear in the document. An anaphor's *antecedent* is one of the previous mentions in the chain. However, there is no agreement on how to select the antecedent for a specific mention. Therefore, we consider only *latent* antecedents, which are selected by the system to be the most likely coreferent predecessing mentions. For example, the system may have higher confidence in linking the last *his* to *Neymar* than in linking it to *the Brazil forward*; *Neymar* then is the latent antecedent of *his*.

The type of coreference we want to resolve is known as the *identity relation*. In the past there has been disagreement on which NP pairs satisfy this relationship. We will apply the restrictions by van Deemter and Kibble (2000) in this thesis. As they point out, some cases of coreference annotated in the MUC corpus (Hirschman and Chin-

[4] From online news article *Napoli back Colombia's Zuniga over Neymar challenge* by Reuters (`http://uk.reuters.com/article/2014/07/08/us-soccer-world-zuniga-napoli-idUKKBN0FD15W20140708`, July 8, 2014).

[5] For example, Lee et al. (2011), the winners of the CoNLL-2011 shared task on coreference resolution (Pradhan et al. 2011), employed this strategy.

chor 1997), a previously popular resource for training and evaluation of resolution systems, are problematic with regards to the definition of coreference. Consider the following examples from van Deemter and Kibble (2000):

(4) a. Whenever *a solution* emerged, we embraced *it*.

b. *Higgins*, once *the president of DD*, is now *a humble university lecturer*.

In (4a), the pronoun *it* acts as a bound anaphor. The *solution* it refers to is generated by the preceding clause's *whenever*. It exists only in the context of this sentence, but is not an actual entity, which is why both mentions are not considered coreferent with each other. This example also illustrates that anaphoricity does not implicate coreference. In the second example, attributes have been marked as coreferent to the attributed entity *Higgins*. While Mitkov (2002, p. 6) deems "noun phrases in copular relation" coreferential as well, van Deemter and Kibble (2000) argue that a substitution test reveals a distortion of the setence's original meaning. Such a test is performed by substituting the possible referent by its antecedent (Mitkov 2002, p. 7), hence the sentence would now read as the nonsensical *Higgins, once Higgins, is now Higgins*. The three mentions are therefore not coreferential. However, attributes in copular or appositive relations establish new information that can be used for subsequent referencing. For example, if there was talk of *the lecturer* later on, we would intuitively link the mention to *Higgins*, as we know that he is a lecturer.

As said in the beginning, resolution systems without semantic knowledge usually fail to resolve *bridging coreference*. The term *bridging* goes back to Clark (1975), who used it to describe utterances in natural languages that require the recipient to draw conclusions to fully comprehend them. It should be noted that he also counts word-identical and pronominal coreference as bridging, as the recipient has to take at least "a millimeter leap" of inference that the two mentions refer to the same entity. Bridging allows a speaker to convey new information in brevity by using given information they can expect to exist in the recipients memory. Consider the following two episodes:

(5) a. Alice chopped down an old oak. She stacked the logs in her garage.

b. Alice chopped down an old oak. **She cut the oak into logs.** She stacked the logs in her garage.

Both sequences convey the same information, yet (5b) is more verbose in doing so. On the other hand, one requires a more elaborate thought process to infer the event of (5b) from (5a). A recipient encountering sequence (5a) first stumbles upon the definiteness of *the logs*, which were not mentioned before, and intuitively searches for an antecedent. He or she can derive from common knowledge that a tree can be cut into logs, and correctly deduces the event written out explicitly in (5b). This is an example of what Clark (1975) calls *reference by association*.

In the scope of coreference resolution, bridging occurs when the head words of an anaphor and its antecedent are not the same (Vieira and Teufel 1997). In (3), a bridging link exists between *Neymar* and *the Brazil forward*. A reader can use three different lines of thought to link the anaphor to *Neymar* and not to *their Colombia defender Juan Zuniga*. First, one might simply know that Neymar plays for the Brazil national soccer team. In this case bridging does not lead to an information gain for the reader. Second, *Brazil/Colombia* as well as *defender/forward* are traits that usually exclude each other, so the correct solution can be found through the process of elimination. Third, after the injury has easily been attributed to *the Brazil forward* based on its syntactic role in the sentence, the possible alternative interpretation of the attacker injuring himself appears unlikely, as in that case he would not have to be backed by his club. The last conclusion requires extensive reasoning over background knowledge which is not available to computer systems for the time being. In contrast, the information required for the first two can be mined automatically from large amounts of text, and we will rely on it in Chapter 7.

Next, we formalize the definition of bridging that can be found in the Berkeley coreference system's source code (Durrett and Klein 2013):

Definition 2. Let Head(m) return the head of mention m and $S = (m_1, \dots, m_n)$ be a non-empty coreference chain. A mention $m_i \in S$ is said to *bridge* if and only if Head(m_i) is a noun and for all $m_j \in S$ with $j < i$ and Head(m_j) being a noun, it holds that Head(m_j) $\neq$ Head(m_i).

Note that if the wrong word is selected as an NP's head, a mention appears bridging to a system when in reality it is not. Also, the definition includes mention pairs which can easily be resolved by looking at other words in the NP besides the head, which is for instance the case with *Lieutenant Cobbs – the lieutenant*. We will exploit this information by means of *attribute* features described in Chapter 7.

In order to solve bridging mentions it is useful to know what semantic relations are used to establish them. Kunz (2010, pp. 92-95) names three common types. The first one is *synonymy*, the near-identity of two words in meaning, e.g. *mother* and *mum*. Either their meaning's connotative or denotative part is identical. The second is *paraphrasing*, wherein word *A* exhibits the property described by word *B* only in the document's context. Paraphrases are the relations being closest to Clark's definition of bridging, as they are often used to convey new information or opinions. For example, a speaker may refer to *an attack* with *the tragedy*, an attribute that is not inherent in the antecedents sense. Kunz also treated *instances*, pairs of proper and common names like *Berlin* and *city*, as paraphrases. We instead assign them to her last group of common relations, which is *hyponymy* resp. *hypernymy*, in which one word acts as another one's class.

Anticipating Section 8.3 in which we examine the bridging types observed in our development set in more detail, we can say that hypernymy is the most common form of bridging. If we remove the above mentioned 'virtual' cases of bridging where the mention phrase already contained all the required information but the system failed to use it, we find that hypernymy relations account for almost 55% of real bridging mentions. Therefore, focusing on them alone should already give a considerable performance boost. The second largest group with 29% is formed by paraphrases. Those are difficult to resolve, as they are often context-dependent, like in the above example of *attack* and *tragedy*. On the other hand, synonymy is only responsible for a mere 4%, which stands in stark contrast to what Kunz (2010, p. 325) observed in a corpus of political essays, where synonyms constituted the second-largest group. This discrepancy may have arisen from the fact that Kunz (2010, p. 92) considered the whole NP for determining the type of semantic relationship that holds between two mentions, while we took only their heads into account.

3 Berkeley Coreference Resolution System

Instead of writing a coreference resolver from scratch, we chose to integrate our distributional features into the Berkeley system by Durrett and Klein (2013). Their objective was to show that sophisticated heuristics are not required to achieve state-of-the-art performance. They designed their supervised resolution system to learn a log-linear model using only shallow lexical and syntactic clues. The model is based on a mention-synchronous instead of a binary classification framework. Both types select a latent (or virtual) antecedent for anaphoric mentions by considering pairwise features of mentions and their antecedent candidates. The binary classification scheme, as for example employed by Soon et al. (2001), first makes a decision about coreference for each individual pair. If none of the individual probabilities of coreference exceed a certain threshold, the mention is deemed being not coreferent with any preceding mention. Otherwise, either the most recent or most probable candidate is selected as the mention's antecedent. In contrast, the mention-synchronous approach, as pursued by Durrett and Klein, additionally learns the likelihood of a mention starting a new entity, which acts like a dynamic threshold.

The simplicity of such a *mention-pair model* comes with a cost, however. As the likelihood of coreference of each mention pair is estimated independently of all others, the coreference relation's transitive property is ignored (Ng 2010). For example, the system might select mention A as the most likely antecedent of B, and B as the most likely one of C, yet A and C disagree in gender and should therefore not end up in the same entity. Furthermore, antecedent candidates are only compared to the current mention, but not to each other, which makes it difficult to learn good weights for continuous feature values (see Section 4.2). Two major alternatives to the mention-pair model have been proposed in the past. *Ranking models* sort the individual feature values of antecedent candidates to select the best one. *Entity-mention models* keep track of the entities they create, and compare unresolved mentions with those clusters instead of their individual members. See Ng (2010) for a discussion on the advantages and disadvantages of each model, Durrett et al. (2013) for an augmentation of the Berkeley system that uses entity-level features, and Rahman and Ng (2009) for a system that combines ranking and entity-mention models.

Let's have a more detailed look at the system by Durrett and Klein (2013). They predict mentions in a high recall, low precision approach, marking all NPs, possessive pronouns and named entities except for those referring to numbers. Classification is performed from left to right, one mention at a time, and without a history. Let m_i denote the current mention we want to resolve. Further, let a_i be a random variable with possible outcomes coming from the set $\{m_1, \ldots, m_i\}$, which is comprised of all previous mentions in the document up to and including the current mention. The values of a_i indicate the latent antecedents mention m_i can be linked to, or in the case of $a_i = m_i$, that the mention either refers to a discourse-new entity or is non-referring. The probability of a_i taking a particular value x given document context d is defined in the usual way of log-linear models:

$$P(x|d) \propto \exp(\mathbf{w}^\top \mathbf{f}(m_i, x, d))$$

with $\mathbf{w}^\top$ being a vector assigning each feature a weight and function $\mathbf{f}(m_i, x, d)$ returning the feature vector for mention m_i and decision x given document context d. If $m_i \neq x$, only pairwise features have non-zero values; in case of equality, features on anaphoricity are the only ones firing. Selecting the best antecedent for mention m_i simply reduces to $\arg\max_{x \in a_i} P(x|d)$. After classification, mentions and their antecedents are put into the same entity. Singletons are removed from the final output.

As mentioned in Chapter 2, in reality there is no such thing as a "single best antecedent" in coreference relationships. For training, Durrett and Klein therefore had to transform entities in the training corpus into examples that are in line with their model. They do so by adding one positive example for each possible antecedent of an anaphor. If a mention is not a member of any entity or has no previous mentions in the coreference chain, an example for a discourse-new ($a_i = m_i$) entity is added instead. All non-coreferent predecessors act as negative examples.

Durrett and Klein trained and evaluated two models on the CoNLL-2011 shared task dataset (cf. Chapter 6). The SURFACE model uses only the following simple features: type of the mention's head (demonstrative, pronominal, common noun, proper noun), sentence/intermediate mentions distance between mention and antecedent, exact string or head match, mention length in words, and lexicalizations of the head, preceding, following, first and last words of a mention. Note that the feature set is rather simple in its nature. Neither syntactic nor semantic features are included. Nevertheless, the SURFACE model significantly outperformed the resolution system by Lee et al. (2011) which won the CoNLL-2011 shared task.

An important part of the system are *feature conjunctions*. If a pairwise feature f fires, two additional features are also activated: $f \wedge type_c$ and $f \wedge type_c \wedge type_a$. The values type$_c$ and type$_a$ depend on the type of the current resp. antecedent mention. If the mention is pronominal, the value equals to the pronoun's citation form, otherwise the type's name ("PROPER" or "NOMINAL") is used. One advantage of feature conjunctions is that animacity, gender and number properties of pronouns are indirectly taken into account. For example, the incompatibility of *president* and *it* can be learned as a negative weight for the feature conjunction $currHead{=}president \wedge curr{=}NOMINAL \wedge ant{=}it$.

As for the reason their system achieved state-of-the-art results with such a simple feature set, the authors found data-driven features to implicitly embed the knowledge one wants to model with the help of more sophisticated heuristics. They even observed a performance decrease when they replaced surface features by their heuristic counterparts. The lexicalizations were able to capture not only a similar, but finer level of detail in comparison to the heuristic features.

While they achieved "easy victories" as stated in their paper's title, Durrett and Klein (2013) considered making good use of semantic features an "uphill battle". They experimented with knowledge from WordNet, named entities already annotated in the corpus, number and gender data for non-pronominals extracted in an unsupervised manner (Bergsma and Lin 2006), and 20 noun clusters obtained from a generative model operating on verb-argument relations and semantic roles (Durrett et al. 2013). While they were able to observe a slight increase in performance, the gain of 0.36 percentage points in average F_1 score on the coreference metrics introduced in Section 8.1 looked not promising to them. Raising the number of resolved bridging mentions decreased the number of correct resolutions for almost all other link types. Thus, in their Final model, Durrett and Klein retained only the number and gender data from the semantic features, and added features on mention nesting, syntactic uni- and bigrams, and the current speaker as prelabeled in the test corpus. Those features altogether had a much higher impact on the performance, increasing the average score by 1.52 percentage points instead.

4 Distributional semantics

As stated in the introduction, our goal was to integrate semantic knowledge into a coreference resolution system without the need of supervised resources. With methods of the field of distributional semantics this knowledge can be retrieved in an unsupervised or semi-supervised fashion.

The origins of distributional semantics go back to Zellig S. Harris' proposal of a distributional method to analyze language structure (Harris 1954). He argued that a speaker's selection of elements in a language is not arbitrary, but rather determined by what has been said and what comes next in an utterance.[6] By comparing the relative frequency of occurrences between elements, it is possible to reveal regularities, a "distributional structure", which Harris believed to be inherent in any language. While his main focus was on discovering morphemes, he also noted a correlation between meaning and co-occurence of words and their surroundings:

> The fact that, for example, not every adjective occurs with every noun can be used as a measure of meaning difference. [... I]f we consider words or morphemes A and B to be more different in meaning than A and C, then we will often find that the distributions of A and B are more different than the distributions of A and C. In other words, difference of meaning correlates with difference of distribution. (Harris 1954, 785 f.)

As Sahlgren (2006, p. 57) points out, this notion of meaning defined entirely by differences is influenced by the structuralism of Ferdinand de Saussure. According to de Saussure (1931/2001, 79 f.; 137 f.), signs (e. g. words, phonemes, etc.) are arbitrary and carry no meaning by themselves. It is what distinguishes a sign from every other that can be described as the sign's meaning. If there was no difference, the sign would have already been superseded by its semantic neighbors. For example, the German pseudo-anglizism *Oldtimer* denotes a veteran car; the word's existence depends on its difference in meaning on the axis of age.

As a consequence of this concept, descriptive classifications like synonymy or antonymy do not automatically apply to the word relationships obtained from distributional methods (Sahlgren 2006, p. 24). Nonetheless they have been tried to acquire similar words and not different ones, following the distributional hypothesis that "words which are similar in meaning occur in similar contexts" (Rubenstein and Goodenough 1965).

Based on Saussure, Sahlgren (2006, pp. 63-67) distinguishes two types of distributional models. A *syntagmatic* model deems signs as similar if they co-occur in the *same* context. An example would be pairs of words in the same sentence, like *dog* and *bark* in the sentence: *the dog barks*. On the other hand, *paradigmatic* models compare signs in *similar* contexts. For example, one might collect all words in object position of the verb *to drink*.

The following sections introduce distributional methods for obtaining semantically similar words. Section 4.1 presents Lin's distributional thesaurus (Lin 1998) which lists for each term the n terms most similar to it. In Section 4.2, we introduce with vector space models another representation of semantic similarity. Section 4.3 describes the foundation for all our semantic features, the JoBimText project (Biemann and Riedl 2013b), a framework that generalizes and refines Lin's thesaurus. Lastly, Section 4.4 outlines a pattern based approach by Hearst (1992) for the automatic acquisition of hyponyms. Pattern based approaches are not considered as distributional (Shi et al. 2010) as the patterns' construction is based upon a descriptive perspective on meaning, but they can be considered related to the distributional hypothesis. In the end, they are syntagmatic in the sense that they match words appearing in the same context, even though the context is limited to a small set of patterns.

4.1 Lin's thesaurus

Lin (1998) created what is today known as a *distributional thesaurus* (DT), a resource that lists for each entry the top n most distributionally similar words. Lin's thesaurus is a paradigmatic model that considers words as similar if they appear in similar syntactic relations. The idea is based on the observation that a verb's set of arguments is semantically restricted (cf. Chapter 1). Consequently, if we encounter a word in the same argument position of that verb frequently, we can expect it to have certain attributes, even if we had never heard of it before. However, not only verbs but all parts of speech within the same sentence can provide valuable semantic information about words they share a grammatical relation with. Consider the following sentence:

(6) I rode the fast ◊.

Assume ◊ to be an unknown word. Without further context, we guess that ◊ is something you can ride, like a bus or a horse. As it is fast, ◊ must also be siginificantly faster than other "ridables", at least if we ignore irony, so a jade (an old, exhausted horse) is ineligible for substitution.

[6] N-gram models exploit this characteristic for predicting the next word in a sentence (see Chapter 4 of Jurafsky and Martin [2014] for an introduction).

To capture a word's context, Lin (1998) collected all its relations from the sentence's dependency parse tree. For example, sentence (6) yields the following tuples of the form *(word, relation type, relational partner)*:[7]

> *(I, subject-of, rode), (rode, has-subject, I), (◊, has-determiner, the), (the, determiner-of, ◊), (◊, has-modifier, fast), (fast, modifier-of, ◊), (rode, has-object, ◊), (◊, object-of, rode).*

Basically, two tuples are generated from each dependency relation, one from the triple *governor – relation type – governee*, and one from its inverse. A word w is then characterized by the individual frequency counts of all tuples matching the pattern $(w,*,*)$. Lin (1998) continues by calculating the pointwise mutual information (PMI) between words w and w' for a given relaton r. The PMI between two outcomes x and y belonging to random variables X and Y is defined as (Manning and Schütze 1999, p. 68):

$$\log \frac{p(x,y)}{p(x)p(y)} \tag{1}$$

We can use maximum likelihood estimation (MLE) to obtain the joint probability of two words occurring in the same relation, as well as the invidual probabilities of words w and w' appearing in their respective positions, by dividing their individual frequency counts by the overall frequency count of that relation observed in a corpus (Hindle 1990):

$$\begin{aligned} p(w,w') &= \frac{\text{count}((w,r,w'))}{\text{count}((*,r,*))}, \\ p(w) &= \frac{\text{count}((w,r,*))}{\text{count}((*,r,*))}, \\ p(w') &= \frac{\text{count}((*,r,w'))}{\text{count}((*,r,*))} \end{aligned}$$

After inserting those probabilities into (1) and applying basic algebra, we get (Lin 1998):

$$\text{PMI}(w,r,w') = \log \frac{\text{count}((w,r,w')) \cdot \text{count}((*,r,*))}{\text{count}((w,r,*)) \cdot \text{count}((*,r,w'))}$$

Lin then defines the similarity between two words w_1 and w_2 as

$$\text{sim}_{Lin}(w_1,w_2) = \frac{\sum_{(r,w')\in T(w_1)\cap T(w_2)}(\text{PMI}(w_1,r,w') + \text{PMI}(w_2,r,w'))}{\sum_{(r,w')\in T(w_1)} \text{PMI}(w_1,r,w') + \sum_{(r,w')\in T(w_2)} \text{PMI}(w_2,r,w')}$$

with $T(w)$ being the set of all pairs (r,w') for which $\text{PMI}(w,r,w')$ is positive. In other words, the similarity is equal to the amount of valuable information of contexts shared between w_1 and w_2, divided by the total amount of valuable information of w_1 and w_2 across all contexts.

To obtain a DT, similarity values are computed between all possible word pairs. For each word, the top n words with the highest values are kept and arranged in descending order. Figure 1 lists for a noun, an adjective and a verb the top ten most similar words in Lin's thesaurus. While it is undeniable that those words are semantically related, there is no clear separation between different types of relationship. Among the first words in the list for *president* are a hypernym (*leader*), a relational antonym (*vice president*), an instance (*Clinton*) and a holonym (*government*). The first three adjectives are antonyms, and the verbs are apparently only related by the fact that their subject argument is usually animate.

Due to this property, DTs are of limited use in the task of coreference resolution. As stated in Chapter 2, bridging is normally realized with synonyms, hypernyms, hyponyms, instances, and in rare cases holonyms and meronyms. An ideal thesaurus would list only those types of semantic relations, and we present an approach to the removal of incompatible words in Section 8.6. Nevertheless, DT are able to provide *negative* evidence. For example, we won't find *chair* among the similar words for *president*. Under the assumption that a system has means of deciding whether a mention is an anaphor, it can use this evidence to rule out "obvious" cases and therefore drastically reduce the number of possible antecedents.

[7] The root relationship is ignored as all verbs can act as roots and therefore no information is gained by including it in the following calculations.

President		**easy**		**(to) work**	
leader	0.264431	difficult	0.278041	do	0.199053
minister	0.251936	hard	0.232852	go to work	0.17571
vice president	0.238359	tough	0.207434	live	0.170713
Clinton	0.238222	simple	0.200408	stay	0.160119
chairman	0.207511	convenient	0.189324	have	0.159123
government	0.206842	impossible	0.177913	make	0.158741
Governor	0.193404	safe	0.177092	play	0.157022
official	0.191428	cheap	0.162851	spend	0.156042
Premier	0.177853	good	0.161774	cooperate	0.154976
Yeltsin	0.173577	important	0.150734	take	0.151423

Figure 1: The ten terms most similar to *president, easy* and *to work* from Lin's thesaurus[8] together with their similarity values.

4.2 Vector space models

As the name suggests, a vector space model (VSM) treats words as vectors in some high-dimensional space. The space may be spanned by any sorts of context features. In its most basic form, a word's extent in a dimension corresponds to its frequency of appearance in the context that represents the dimension. Figure 2 shows a syntagmatic VSM with a two-dimensional vector space, in which each dimension represents a different document.

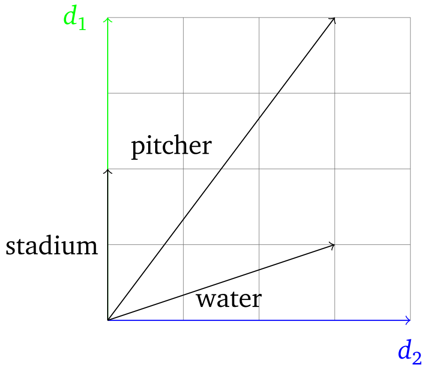


Figure 2: Vector space created from two documents with a vocabulary of three different terms. A word's extent in one dimension represents the number of times it occurred in the associated document.

The semantic similarity of two word vectors u and v in an n-dimensional space can be computed by means of various measures, e.g. cosine similarity (Manning and Schütze 1999, p. 541):

$$\cos(u, v) = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}} \tag{2}$$

Syntagmatic VSMs tend to cluster words if they originate from the same semantic field. This property has earned them popularity among the information retrieval community, where different documents represent different topics, and query term vectors therefore match documents of interest (Sahlgren 2006, p. 65). However, in single document coreference resolution, this may lead to many non-coreferent mentions considered semantically similar by the syntagmatic VSM. For example, *parliament* and *law* can never refer to the same entity, but co-occurs frequently in many documents and would therefore get a high similarity value.

As said in Chapter 2, the coreference of two mentions can be established by substituting one by the other. From this, one can deduce that a mention can always appear in all contexts of all mentions in its coreference chain. Thus we favor a similarity measure that assigns high values to words that appear in similar contexts; hence, a paradigmatic model.

[8] `http://webdocs.cs.ualberta.ca/~lindek/Downloads/sim.tgz`

Simple paradigmatic VSMs use a *context window*, a word sequence of fixed size, with co-occurring words as context features (cf. Sahlgren 2006, 67 f.). Syntax-based VSMs construct a semantic space based on sentence parses. Padó and Lapata (2007) noted that the dependency relation tuples (w, r, w') collected by Lin (1998) during thesaurus construction basically span a vector space, with each word w being a vector and all (r, w') observed in the training corpus its dimensions. They generalized this idea by considering entire paths in a dependency parse and introducing a function that maps paths to dimensions, which also allows to build conventional word × word co-occurence spaces. The structured VSM by Erk and Padó (2008) treats words as tuples consisting of the word w itself and one vector for each dependency relation r the word was observed in. Those vectors are filled with the partner words w' from the corresponding relation. The model can then be used to compute semantic similarity in context by disambiguating the target word sense using selectional preference information from the tuples.

When using a ranking model for coreference resolution, the integration of a VSM is straightforward. Let ϕ be a similarity measure that takes two vectors of the VSM's space as arguments, and A the set of possible antecedents of a mention m. Then the antecedent $a \in A$ with the highest similarity score $\phi(\text{head}(a), \text{head}(m))$ is selected as the antecedent of m (Poesio et al. 1998).[9] However, a mention-pair model like the Berkeley system scores pairs of mentions independently. We are no longer ranking $|A|$ antecedents, but are prompted with the question whether two words u and v are semantically similar enough to be considered coreferent. A simple solution would be to use the raw or binned similarity score as the feature value. This way, the system could learn weights to normalize the score's distribution and to learn its relevance in comparison to all other features. Yet, as Lee (1999) points out, substitutability is a non-symmetric property: e.g., the word *city* may be the best substitute for *Paris*, but not vice versa. A measure like cosine similarity is symmetrical, thus averaging over the likelihood of both u substituting v and v substituting u. Iterating over the complete vector space to obtain a relative score is computationally infeasible. The problem can be solved by applying a directional similarity measure (Kotlerman et al. 2010).

However, by converting the vector space to a DT, we can keep the measure symmetric. While its construction requires a more computer intensive procedure than building a VSM, there exist techniques using approximation or pruning to speed up the computation significantly (Curran and Moens 2002; Riedl and Biemann 2013). These methods inevitably result in losing the ability to compute the original score of *all* word pairs, but they are designed in a way such that the lost pairs are likely dissimilar, and thus not required in our task. Once constructed, the question whether u is the most similar term to v can be answered in constant time by looking up u in v's similarity list.

4.3 The JoBimText Framework

The	**quick**	**brown**	**fox**	**jumps**	**over**	**the**	**lazy**	**dog.**
a	swift	gray	coyote	leaps		a	stupid	cat
another	fast	red	raccoon	climbs		another	arrogant	pet
...	...	...	...	...		...	...	...

Figure 3: Two-dimensional text (based on Biemann and Riedl 2013b, p. 57) using an actual DT. The example also shows the difference between surface words and terms, as the word *over* is only considered as a context feature and not as a term by the underlying dependency parse holing system.

Biemann and Riedl (2013b) introduce the notion of two-dimensional text as a metaphor for semantic analysis. In addition to the sequential, syntagmatic layer constituted by the text itself, there exists a paradigmatic dimension of hidden meaning. Each language element in the text has an *expansion*, an ordered list of semantically similar signs. Figure 3 shows an example of two-dimensional text.

The expansions are filled using a DT. They abstract away from Lin's method by creating a framework that operates on arbitrary pairs of *terms* and *context features*, whereby the same term can exhibit multiple (including zero) features. This way, DTs can expand other structures than words, like multi-word expressions, phrases, or whole sentences. Similarly, the context is not restricted to dependency relations, but could be represented by N-grams, nearest content words, or others. The initial acquisition of terms and their contexts, which Biemann and Riedl (2013b) call the *holing operation*, is the only step in the DT's computation that has to be informed about the nature of the context features.[10]

[9] This example assumes semantic similarity to be the only feature used during resolution. In case of multiple features, weighting and/or mapping the similarity to its rank among all antecedents A is preferred.

[10] The holing operation is also required later on for expansion, as we need to know how terms are constructed if we want to look them up in the DT.

The holing (or "@") operation transforms a text into a sequence of terms and yields for every term a set of pairs in the form *(term, context feature)*. A holing symbol ("@") takes the place of the term in its features to capture positional information. For example, assume we have a holing operation based on bigrams that uses lemmatized words in lowercase as its terms and their direct neighbors as their respective context features. If we run it on the text *I rode the fastest horse*, the @-operation returns the following pairs: *(I, @ ride), (ride, I @), (ride, @ the), (the, ride @), (the, @ fast), (horse, fast @)*.

It should be noted that the @-operation is not restricted to a single hole. A term is actually a tuple, and holes are indexed so that in the original context the i^{th} hole was filled with the i^{th} element from the term tuple, thus *(I, @ ride)* is actually *((I), $@_1$ ride)*. As we did not use DTs with multiple holes in our work, we will stick with the simple representation from above for the remainder of this thesis.

To compute a DT from term-feature pairs, Biemann and Riedl (2013b) proceeded as Lin did, with two main differences. The first one is the size of the corpus; while Lin mined 64 million words, they drew the pairs from 120 million sentences. Riedl and Biemann (2013) showed that a DT's quality increases with the size of the corpus it was computed on. However, to be able to compute the DT on data this large in reasonable time, Biemann and Riedl (2013b) apply pruning by keeping only the p context features with the highest significance value per term. Instead of using Lin's measure, they chose Local Mutual Information (LMI; Evert 2005, p. 89) to compute the significance of a term-context pair. LMI is calculated as

$$\text{LMI}(\text{term}, \text{feature}) = \text{count}(\text{term}, \text{feature}) \cdot \log_2(\frac{\text{count}(\text{term}, \text{feature})}{\text{count}(\text{term})\,\text{count}(\text{feature})})$$

Their second change to Lin's method is the way how similar terms are being clustered. Lin's similarity measure requires a pairwise comparison of all terms resulting in a time complexity quadratic to the number of terms, which is undesirable when handling large corpora. In contrast, Biemann and Riedl (2013b) first aggregate terms per feature and compare two terms only if they have at least one feature in common. The similarity value of two terms t_1 and t_2 simply equals to the number of shared features, or, formally, $|\mathit{features}(t_1) \cap \mathit{features}(t_2)|$, with *features(·)* returning all features which survived the pruning step for a given term.

Like Lin, they retain only the n most similar terms per term. However, they also store the term-context pairs, which allows them to propose a unique method which we will refer to as *context-sensitive re-ranking* in this thesis. It is motivated by the lexical substitution task in which a system has to suggest (or choose from a list) a substitute for a word that conserves the syntax and meaning of the original sentence as close as possible (McCarthy and Navigli 2009). The method aims to favor those words in the expansion that are likely to appear in the given context, which is especially useful in the case of words with mutiple senses and homonyms.

Let us exemplify their method by applying it to the following example sentence:

(7) The dangerous mole infiltrated Buckingham Palace.

We would like to substitute the underlined word *mole*, which is made difficult by the fact that it could refer to an animal, a lesion, or a spy. The context however suggests the last one to be most fitting.

First, the term corresponding to *mole* is looked up in the DT as normal, producing what is called a *prior expansion* (Gliozzo et al. 2013), which is the precomputed list of n similar words. Assume our DT with $n = 8$ returns the following list: *lesion, scar, rabbit, freckle, squirrel, spy, bruise, informant*. Next, Biemann and Riedl (2013b) calculate for each term t the likelihood of it appearing in the context of the substituted word. Let C be the set of context features for the to-be-substituted word, T its prior expansion, and $\mathit{sig}(t,c)$ return the significance score for a term-context pair (t,c) stored in the DT model, or zero if this pair does not exist. The context score of a term $t \in T$ is then calculated as the harmonic mean of the individual $\mathit{sig}(t,c)$ scores for each $c \in C$. Plus-one-smoothing is applied to the term-context scores so that terms which co-occurred only with some context features are kept in the final result. Additionally, we adapt the term frequency weighting scheme from Biemann and Riedl (2013a) to decrease the score returned by *sig* for features that contain common terms, which are deemed uninformative. For this, a function *term* is defined that extracts a single term from a context feature.[11]

The resulting formula reads as follows:

$$\text{rank}(t) = \frac{\sum_{c \in C} \text{count}(\text{term}(c))^{-1}}{\sum_{c \in C} (\text{count}(\text{term}(c))\,\text{sig}(t,c) + 1)^{-1}} \tag{3}$$

[11] Clearly this scheme is only applicable if the holing systems uses terms in context features in the first place, which is for example the case in a dependency based thesaurus. In other cases, the frequency of the features themselves might be a better choice.

The terms from the prior expansion T are re-ranked in descending order of their context scores to obtain the final, context-sensitive expansion.

Returning to the example and assuming a dependency parse holing system, the above sentence yields the context features *dangerous modifier-of @* and *@ subject-of infiltrate*. We might, for example, obtain a significance score of 120 for the pair *(spy, @ subject-of infiltrate)*, but a score of 0 for *(squirrel, dangerous modifier-of @)*. After applying equation (3) to each of the prior expansion's eight terms and sorting them, we acquire a list in which the top terms better suit the subject at hand: *spy, informant, lesion, rabbit, scar, freckle, squirrel, bruise*.

We will employ context-sensitive re-ranking as a feature for coreference resolution in Section 7.1 and discuss their impact on coreference resolution in Section 8.2. Precomputed models and a framework for computation and lexical expansion are available in form of the *JoBimText framework*[12] (Biemann and Riedl 2013b; Gliozzo et al. 2013), which we made use of to compute our distributional features. See Chapter 6 for details on the model parameters.

4.4 Hearst patterns

The online encyclopedia Wikipedia states in its guidelines on writing an article's introduction that "[t]he first sentence should tell the nonspecialist reader what (or who) the subject is".[13] Thus most Wikipedia articles start in a fashion similar to the following first sentence on grapefruit:[14]

(8) The grapefruit (Citrus × paradisi) is a subtropical citrus tree known for its sour to semi-sweet fruit, an 18th-century hybrid first bred in Barbados.

One can extract the information that *grapefruit* is a member of the class of *subtropical citrus trees* from this sentence by simple pattern matching using the pattern "X *is a(n)* Y". Under the assumption that all Wikipedia articles are prefaced alike, we would be able to create a large thesaurus of hyponymous words just by looking at the first paragraph. Hearst (1992) was the first to employ patterns for the automatic extraction of hyponyms from raw text. Instead of the above pattern, she used a small set of enumerations, which are depicted in Figure 4. Since they contain a repeating (Kleene star) expression, multiple hyponyms may be acquired in a single match.

1. NP such as (NP,)* (and|or) NP
   A variety of board games such as chess, checkers or Halma are playable.
2. NP (, NP)* (and|or) other NP
   Dogs, cats and other pets ...
3. such NP as (NP,)* (and|or) NP
   ... preferred by such leaders as Barack Obama, Angela Merkel and François Hollande.
4. NP (including|especially) (NP ,)* (or | and) NP
   ... periodicals including scientific journals and tabloid magazines ...

Figure 4: Patterns used by Hearst (1992) for the acquisition of hyponym-hypernym relations, each acommpanied by an example. Optional commas are omitted for clarity.

Despite their simplicity, those patterns produce relations of high quality, as Hearst's focus was on precision. On the downside, their relative sparseness in comparison to the amount of documents in which hyponyms co-occur with their hypernyms leads to low recall, but this effect can be alleviated by using a larger corpus. Instead of the full dependency parse required to create Lin's thesaurus, a constituent analyzer is sufficient in detecting NPs, so the approach can be adopted to any langauge for which such a tool exists. The only difficult task is the definition of new patterns, though those can be found in a bootstrapping approach using an initial seed of known hyponym–hypernym pairs (Hearst 1992).

Hearst patterns are sometimes called IS-A patterns, named after the relation holding between the words they find. We will refer to all patterns that reveal words in a linguistic relation as "Hearst patterns" during the rest of this work, and apply the term "IS-A pattern" only to patterns of the hypernymic kind.

Hearst also noted that some of the mined pairs do not express a hypernymic relationship, but are rather based on opinion or the surrounding context, such as *Washington* IS-A *nationalist* or *aircraft* IS-A *target*. While this might be

[12] http://www.jobimtext.org/
[13] http://en.wikipedia.org/w/index.php?title=Wikipedia:Manual_of_Style/Lead_section&oldid=617138662
[14] http://en.wikipedia.org/w/index.php?title=Grapefruit&oldid=614759908

a problem when the goal is to extend or create a dictionary, we consider those "CAN-BE" relations as information of equal importance for resolving bridging mentions. Not only may they appear in a similar context or be expressed by a speaker with the same opinion, but they may also be a realization of the paraphrase relation introduced in Chapter 2.

The IS-A patterns' second disadvantage is their lack of any taxonomy. Although we may learn that *cat IS-A animal* and *cat IS-A feline*, we do not know that *feline* is a more specific hypernym than *animal*. While there exist methods to create a hyponym hierarchy (see Caraballo and Charniak 1999 for details), we consider a distinction superfluous for the task at hand.

Pattern-based approaches to the acquisition of all kinds of information are popular among researchers in the field of coreference resolution. We introduce some of them in the following section. We use Hearst-style patterns to acquire hyponyms in Section 7.2 and incompatible words in Section 8.6.

5 Related work

In this section we survey previous attempts to acquire semantic knowledge for coreference resolution, while we confine ourselves to the two aspects of coreference we want to address: bridging mentions and selectional restrictions.

5.1 Resolving bridging mentions

A wide range of works on coreference resolution proposed methods to resolve bridging mentions, either as the main focus of the research or en passant while presenting a novel resolution system. We first discuss the advantages and disadvantages of drawing semantic information from lexical resources made by humans. Following this, we present previous approaches to automatically extract this knowledge from unlabeled corpora.

5.1.1 Using manually created lexical resources

To this day, WordNet (Miller 1995) is a popular knowledge resource in the field of coreference resolution. It arranges words in a graph whose arcs are labeled with hyponymic and synonymic relations holding between them, facilitating antecedent selection for those types of bridging mentions. Moreover, as the database is crafted by lexicographers,[15] the information it contains is of high quality.

Various methods have been proposed to make WordNet data available as features to a resolution system. Vieira and Teufel (1997) tested whether two words are neighbors in the WordNet graph, Poesio et al. (2004) obtained a similarity measure by normalizing the distance of the shortest path between two words by the graph's diameter, and Soon et al. (2001) determined the semantic class of a mention by walking up its head's hyponym hierarchy until a word from a hand-written list of semantic classes is reached.

Among the many participants who used WordNet in the CoNLL-2011 shared task on unrestricted coreference (Pradhan et al. 2011), two of them, Lee et al. (2011) and Zhou et al. (2011), found semantic features based on it to be detrimental to their system's performance.

Ponzetto and Strube (2006) evaluated the use of Wikipedia as an alternative source of semantic knowledge. They implemented features based on Wikipedia's category system, as well as cross-references and the textual overlap between two articles. They found Wikipedia features, while helping to improve over the baseline, to be inferior compared to ones based on WordNet. A combination of both yielded the best results.

Manually crafted resources have some shortcomings, some of which were outlined in Chapter 1. Not only are they expensive to create, but they are also too small. In their detailed study of techniques for the resolution of definite nouns, Vieira and Poesio (2000, pp. 572-574) noted that most missing links involved synonyms. These errors were caused by missing entries, as well as spelling differences (e.g. *spinoff* vs. *spin-off*) and domain-specific senses not covered by WordNet. As opposed to this, a DT can be trained on a corpus containing texts of a specific domain (cf. Gasperin et al. 2004) and also conveniently lists common spelling differences (including errors), as those total synonyms are distributed quite similarly. For example, *spin-off* is the top word in the expansion of *spinoff* and vice versa in the DT introduced in Chapter 6.

5.1.2 Using data obtained from unannotated corpora

Pattern-based approaches to coreference resolution usually extract pairs of coreferent words directly. Kobdani et al. (2011) pursued a self-training approach. They identified possible coreferent pairs in an unlabeled corpus using strong heuristics and an association measure (pairwise mutual information) in an unsupervised fashion, and then trained a supervised system on these pairs. Haghighi and Klein (2009) first mined a small set of compatible heads from appositive and predicative-nominative relations. These guessed pairs were then used as seeds in a bootstrap approach to obtain Hearst patterns of coreference, which in turn allow to find even more pairs of coreferent NP heads. As Raghunathan (2010, pp. 14-18) points out, this method produces a lot of noisy seeds which give rise to imprecise patterns. He tried to remove bad seeds using Lin's thesaurus, but found the DT to be too noisy to base coreference decisions on it. Eventually, he employed WordNet for seed validation.

Moving on to distributional models, Poesio et al. (1998) employed a VSM to cluster similar words and experimented with cosine, manhattan and euclidean distance as similarity measures. They chose the top 2,000 common content words in a small-sized window (up to 30 words) as the space's dimensions. While they were able to obtain better results than with a WordNet-based approach, selecting the semantically most similar antecedent was responsible for 40% of the wrong links. Gasperin et al. (2004) built two DTs, one for the French and one for the

[15] http://wordnet.princeton.edu/wordnet/frequently-asked-questions/database/

Portuguese language, using dependency relations as context features and the Jaccard index as the similarity measure. Only the 15 most similar words per term were kept. While the baseline which selected always the most recent antecedent was stronger than the DT-based method on French texts, using the Portuguese DT prove successful by achieving a 21% F_1 score on the resolution of bridging mentions. Versley (2007) tested multiple distributional and pattern-based similarity measures individually and in combination to resolve bridging mentions in German texts. Additionally, he compared the results with GermaNet (a German version of WordNet) and tried to combine this resource with the other features as well. Among the features were a single German IS-A pattern, Lin's thesaurus, a VSM based on weighted dependency relations, and pairs of distance-based coreference guesses. When testing each feature separately, the features based on distributional similarity showed the highest recall, but a lookup of hypernymy in GermaNet outperformed all others in F measure.

We would have liked to compare our results on bridging mentions to those of Poesio et al. (1998), Gasperin et al. (2004) and Versley (2007), but two main differences between our and their experimental setup prevented us from doing so. The first one is that their systems are rule-based and use no additional features beyond the semantic heuristics. In contrast, we extended a fully-fledged coreference resolution system that, being a log linear model, is also able to weight each single semantic feature and thus take their mutual dependence into account. Secondly, all three of them were evaluated on bridging mentions whose anaphoricity had been made transparent to the system. Therefore the only error possible error was the selection of the wrong antecedent. We explore the utility of semantic features in a predicted mention setting, in which the system has to determine whether a mention is anaphoric by itself. This gives rise to superfluous mentions and entities being annotated as well if the semantic features are imprecise or appear too convincing to the system.

Distributional and pattern-based features were integrated by Ng (2007) into the machine learner by Soon et al. (2001). He obtained the patterns in a similar way to Haghighi and Klein (2009), but estimated their accuracy by running them over an annotated corpus, and used them directly as a feature instead of mining coreferent pairs from unlabeled data first. Ng also used Lin's DT to create a semantic similarity feature that fired if a mention's head was among the top 5 words in the other mention's head's expansion, or vice versa. In their entity clustering model, Lee et al. (2012) computed semantic similarity between two clusters by retrieving the 10 most similar words from Lin's thesaurus for each word in each of the clusters, converting those lists to two word vectors, one for each cluster, and taking their cosine similarity.

Recasens et al. (2013) made use of the fact that different articles covering the same news story have similar, but not identical wording. For example, one text may contain the sentence *the John Doe concert had to be canceled* and another one *the singer's concert was canceled*, which allows to infer that the underlined phrases are mutually substitutable. This requires a corpus of aligned news articles, which they obtained from an automatic news aggregator. Like with our approach, they noticed a significant decrease in precision, but a much higher increase in recall.

5.2 Modeling selectional restrictions

Already Harris (1954, p. 197) cited selectional restrictions as an example of the distributional property of language. The majority of approaches to selectional restriction or selectional preference is driven by comparisons of distributions. Presented with a single verb and multiple possible noun arguments, they select the one that was observed most often in argument position of that particular verb in a training corpus.

The idea of modeling selectional preference for coreference resolution with the help of knowledge mined from large amounts of raw text dates back to Dagan and Itai (1990). They acquired statistics on pairs of nouns and their partners from relations in which they were governed by a verb or were modified by an adjective. During resolution, they enforced selectional restrictions only if a pronoun has more than one possible antecedent that agrees with it in number, gender and animacity. If the anaphor partakes in one of the relations above, they selected the antecedent whose head was observed most frequent in the same argument position in the training corpus.

Yang et al. (2005) made some small changes to the method by Dagan and Itai. They replaced the adjective by a possessive relation that captures patterns like *power poss-of government*. To handle data sparseness, proper noun phrases are reduced to their NEs as labeled by a named entity recognizer. The antecedent selection is based on an MLE estimate instead of raw counts to compensate for nouns that would be preferred just because they are more frequent than others in general. Additionally, they submitted queries in which the anaphor was substituted by the candidate to a web search engine and calculated the MLE estimate based on the number of returned results. We adapt the use of MLE for measuring selectional preference when we introduce our context-based DT expansion in Section 7.3.

Bean and Riloff (2004) extracted *contextual role knowledge* from plain text to build their BABAR resolution system. In speech, nouns play thematic roles determined by their context. For example, *Columbus* plays the role of an explorer in the sentence *Columbus discovered America*. They automatically created patterns for every verb-argument structure observed in a corpus, normalized them, and applied them to the same corpus to retrieve valid arguments for verbs. At resolution time, a mention and its possible antecedent were tested whether they could appear in each other's context. Bean and Riloff also clustered patterns which co-occur with the same entity, e.g. *X was killed* followed by *the murder of X*. To train those on coreferent mentions without requiring a labeled corpus, they used only 'safe guesses' of coreference like recurring proper nouns or appositve relations.

It is unlikely that we observe a noun in all valid argument positions, especially if it is infrequent (Picard 1999). Therefore many works apply smoothing to generalize their models to unseen verb-argument tuples. Resnik (1996) built a probabilistic model around hypernyms of arguments, which he extracted from WordNet, and which acted as conceptual classes (like BEVERAGE or OBJECT). Bergsma et al. (2008a) replaced the semantic classes by word clusters obtained in an unsupervised fashion. Other smoothing techniques consider the distributional similarity of arguments: Both syntax-based DTs (Erk 2007; Calvo et al. 2009) and VSMs (Erk et al. 2010) have been used successfully in the task.

Our approach to modeling selectional restrictions by expanding the context (cf. Section 7.3) is very similar to verb argument expectations by Lenci (2011). He, too, lists the words that are most likely to appear as arguments of a verb. However, he uses a structured VSM, while we relied on a distributional thesaurus. Furthermore, when determining the preference of an argument, he also takes the governing verb's other arguments into account, e.g. to distinguish between *the journalist checks X* and *the mechanic checks X*. In contrast, our method does not make any statement about the type of features. While the DT we use in this work captures only the immediate neighbors in the dependency tree as context features, it is possible to replace the holing operation by a more fine-grained one without modifying the expansion scheme. Moreover, we consider all relations of a NP head, not only those that interact with a verb. This allows us to better guess the properties of an out-of-vocabulary word.

6 Experimental setup

For training and testing the coreference system, we used data made available in the context of the CoNLL-2011 shared task on coreference (Pradhan et al. 2011). It contains the English section of the OntoNotes v4.0 corpus (Hovy et al. 2006), annotated with syntactical, word sense, named entity, speaker, semantic role, and coreference information. The corpus has a size of 1,3M words distributed over 2,999 documents, which makes it larger than the two hitherto most important corpora for English coreference, MUC and ACE, combined. It includes texts from newscast, newswire, magazines, telephone and web communications (cf. Pradhan et al. 2011).

We adopted the splits of training, development and test data defined by the task organizers. We limited our experiments to the documents' AUTO versions, where all annoations except those for coreference and speaker identification were produced by automated, state-of-the-art tools (Pradhan et al. 2011). The system was told to ignore gold mention boundaries and to predict them instead during both training and evaluation. Additionally, we provided the system with distributionally obtained gender and number data by Bergsma and Lin (2006) from the task's website[16], as it is employed by the Berkeley FINAL system.

A special type of coreference annotated in the CoNLL-2011 shared task is event coreference. It occurs when a verb expressing an event is referred to by another verb or a noun, like in the following example (Pradhan et al. 2011):

(9) Sales of passenger cars **grew** 22%. **The strong growth** followed year-to-year increases.

In this thesis, we are only concerned with identity coreference as defined in Chapter 2, in which we restricted the set of mentions to noun phrases. Thus, like the Berkeley system, we ignore the phenomenon of event coreference. This puts an upper bound on the system's performance, yet the impact is marginal. Only 9% of the gold mentions in the corpus are verbs (Pradhan et al. 2011).

Although the JoBimText framework allows arbitrary holing systems, our main focus was on a DT based on dependency parsing.[17] We used a DT from the project's homepage[18] which was computed on 120M sentences of English news articles taken from the Leipzig Corpora Collection (Richter et al. 2006) and the Gigaword corpus (Parker et al. 2011). The number p of context features for calculating term similarity was 1000. In this DT's holing system, terms are pairs comprised of a single word's base form as produced by ASV Toolbox (Biemann et al. 2008) and its POS tag. Context features are triples extracted from collapsed dependency parses by the Stanford parser (Marneffe et al. 2006) containing the partner word in term form and the relation type as labeled by the parser. For example, one of the term-context pairs obtained from the sentence *the actors left* is *(<actor,NN>, @ subj-of <leave,VB>)*.

The DT comes with an additional layer with its terms organized in sense clusters produced by the Chinese Whispers algorithm (Biemann 2006), allowing word sense disambiguation using context features. The clusters are enriched with aggregated hypernyms from IS-A patterns, which were applied to the source corpus (Gliozzo et al. 2013). The aggregation helps to compensate for the sparseness of Hearst patterns, but inevitably reduces precision; for example, the DT lists *bird* as one of the clustered hypernyms for *cat*.

To apply the holing operation to the training corpus documents, we translated the Penn-Treebank syntax trees into dependency parses using the the Stanford parser's dedicated conversion function (Marneffe et al. 2006) and converted the POS labels to their notation as used in the DT. For the sake of consistency, ASV Toolbox was employed for our lemmatization as well.

[16] http://conll.cemantix.org/2011/

[17] See Section 8.7 for a comparison with a context window holing operation.

[18] http://sourceforge.net/p/jobimtext/wiki/models/

7 Feature design

The Berkeley system distinguishes between two types of features: features which fire on single mentions and which are used to determine whether a mention is anaphoric, and pairwise features that compare a current mention *cur* with its potential antecedent *ant*. Our set of distributional features consisted solely of the pairwise kind. In Section 7.1 and Section 7.3 we present features based on distributional similarity, while Section 7.2 describes our integration of hypernym relation information extracted with IS-A patterns.

We tried to create features that are applicable independently of the DT's underlying holing operation. We defined two functions term(·) and context(·) which map a mention to its term representation or its set of context features, respectively, according to the DT's holing system. For DTs with single words as terms, the functions will operate on mention heads, e.g. within an N-gram system, *term([the blue sky])* yields *sky*. However, our main focus was on the dependency thesaurus, and some features do not translate well to other holing systems. For example, our context-based expansion (see Section 7.3) maps pronouns to the first common or proper noun, information which is available in our main DT whose terms contain the word's part of speech, but not in an expansion produced by a DT with a simple N-gram holing system. Such a system would have to use a list of English pronouns and demonstratives to achieve the same effect.

Since the Berkeley system relies on boolean features, despite being a log-linear model, we had to discretize continuous feature values. For features that return ranks, we decided to simply partition the expansion size of 200 into 10 equal-width intervals and added a feature for each of them whose value is true if the interval contains the value returned by the corresponding ranking function. We allowed the system a little bit of fine tuning by using the raw value if contained in the interval $[-2, 20]$, whereby negative values indicate that the target term's expansion was either empty or did not contain the query term.[19] Proportional real values from the interval $[0, 1]$ were rounded to the first decimal digit.

Most of our features are asymmetrical and herein described by referring to a pair of mentions m_1 and m_2. We add a separate feature for each of the two possible allocations: $m_1 = cur \wedge m_2 = ant$ and $m_1 = ant \wedge m_2 = cur$, and also capture the allocation so the system can learn different weights based on the direction of linkage, which can be a useful clue. For example, in coreference chains involving IS-A relations, a hypernym is more likely to follow a hyponym than vice versa (Kunz 2010, p. 325).

7.1 Word-based expansion

Our first feature, PRIOR, is based on the DT's prior expansions containing the most similar terms per term. While Ng (2007) used a boolean feature that returned true for a pair of mentions NP_1 and NP_2 if NP_1 was among the top five expansion words of NP_2 or vice versa, we considered the whole (200 terms) expansion and used the rank instead of a simple check whether a word is contained. Compared to the top five ranks, we found twice as many heads of coreferent mentions among the remaining 195 entries. Being a log linear model instead of a decision tree classifier like in Ng (2007), the Berkeley system should be able to learn weights from the explicit ranks that reflect the grade of similarity between two terms. We defined a function *prior*(·, ·) as follows:

$$\text{prior}(t_1, t_2) = \begin{cases} 0 & \text{if } t_1 = t_2 \\ -2 & \text{if expansion}(t_2) = \emptyset \\ -1 & \text{if } t_1 \notin \text{expansion}(t_2) \\ \text{position}(t_1, \text{expansion}(t_2)) & \text{otherwise} \end{cases}$$

where t_1 and t_2 are terms extracted from mentions in the way described above. We added a feature with the value of *prior(term(m_1), term(m_2))* if neither mention's head was a pronoun or demonstrative. Figure 5 shows a toy example case in which the PRIOR feature helps resolving a bridging mention.

An absolute rank is an inaccurate measure of similarity. A rank of ten, for example, has a different importance depending on the expansion's size and the other similar terms' meaning. By assigning weights to ranks, the system averages over the similarity values assigned to each single rank. We therefore also calculated the expansion overlap between two terms as

$$\text{overlap}(t_1, t_2) = \frac{|\,\text{expansion}(t_1) \cap \text{expansion}(t_2)|}{\min(|\,\text{expansion}(t_1)|, |\,\text{expansion}(t_2)|)}$$

[19] We also experimented with supervised binning using the ChiMerge algorithm (Kerber 1992), with non-anaphoric/anaphoric mentions with no head match as the two classes and p=.95, but noticed no difference in performance.

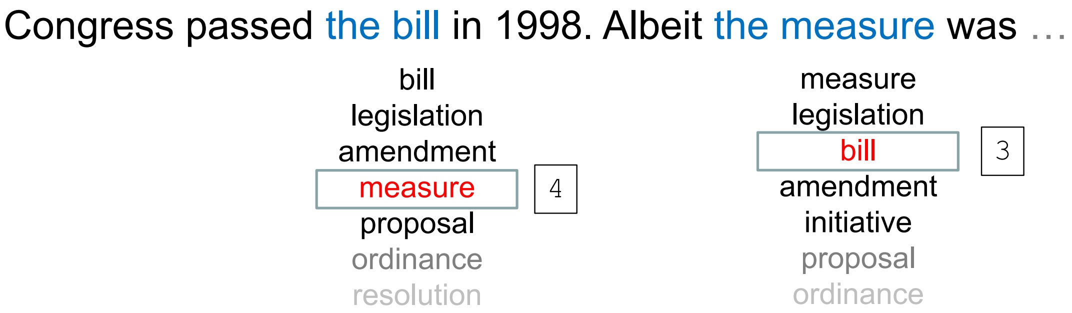


Figure 5: PRIOR feature: Looking up the current mention's head in the candidate antecedent head's expansion, and vice versa.

and added the discretized results as SHARED_PRIOR feature values.

We also wanted to capture entity attributes introduced in the discourse. Consider the following example:

(10) [Doctor Jane Doe] won the Nobel Prize in physics. [The scientist] discovered a correlation between ...

Clearly, the mentions *[the scientist]* and *[doctor Jane Doe]* belong to the same entity, as the former references an attribute of the latter. Our head-centric features however are unable to exploit the fact that *doctor* and *scientist* are similar words. Thus we created a feature ATTR_PRIOR for which we expand term(m_2) and look up each attribute of m_1. Inspired by Vieira and Poesio (2000, 556 f.; 560), we considered as attributes all partners in a dependency relation with the head noun of mention m_1, provided the relation is of one of the following types: copula (*Jane is a scientist*), apposition (*Jane, a scientist*), relative clause (*Jane, who is a scientist*) or compound (*the scientist Jane*). Each of them is converted to a term and searched for in the expansion of term(m_2). The highest rank is reported as the feature's value. We added separate feature values if none of the attributes was contained, or if the prior expansion was empty.

Context-sensitive re-ranking was performed as described in Section 4.3 and reported as a rank-based feature named RERANK. Like Biemann and Riedl (2013b), we reduced the set of context features by keeping only dependency relations with content words, i.e. adjectives, adverbs, nouns and verbs. A stable sorting algorithm is employed to fall back to the initial order in case of ties. We look up *term(m_1)* in *term(m_2)*'s re-ranked expansion and determine the feature value as above for PRIOR. As the DT we used always had the expanded term itself at the first position of its prior expansion, we moved it to the top in the re-ranked expansion as well to ensure symmetry. After all, a term should be its own most similar term regardless of context.

We observed that the Berkeley system's feature conjunctions help taking into account that the prior expansion's quality varies greatly depending on the expanded term's part of speech. The case that a common noun was ranked first in another common noun's prior expansion was assigned a much larger weight than the same situation involving two proper nouns. This makes up for the fact that expansions of proper nouns are dominated by other, cohyponymous proper nouns.

We tried to simulate a FrameNet feature by Rahman and Ng (2011), which exploits the fact that mentions occupying the same argument position of verbs from the same frame are likely coreferent. They labeled mention heads with semantic roles and checked if the predicates of candidate pairs are in the same frame in FrameNet. They created a feature for each element in the cartesian product between the two semantic roles and the outcome of the FrameNet test. For our ROLE feature we chose to distinguish only between subject and object positions as labeled by the dependency parser. The argument combinations are reduced to a boolean value that is true if the roles of both mentions match. Instead of a lookup in FrameNet, we expanded the verb governing the first mention and retrieved the rank for the second mention's governing verb. Thus, unlike in Rahman and Ng (2011), our feature is asymmetric. We discretized and swapped the two mentions' roles as usual. We hoped that the DT's large size in comparison to FrameNet would reveal beneficial correlations. Instead, the feature actually had a negative impact on the system's performance. We offer two possible explanations for this phenomenon. First, we consider verbs to be among the parts of speech with the weakest expansion, as can bee seen in Figure 1 from Section 4.1. They were chosen as a dependency parse's root because predicates play the most important role in a sentence when it comes to meaning. This makes verbs very strong context features, but their own distributional similarity is computed mostly on their arguments, which may origin from very different semantic fields. A second explanation is that this coreference phenomenon might be rather rare and many non-coreferent parallelisms exist. As a result, we refrained from using the ROLE feature in further experiments.

7.2 IS-A patterns

To address bridges involving hyponymy, we defined three features that make use of IS-A patterns, named IS-IS-A, ATTR_IS-IS-A, and SHARED_IS-As. They all operate only on pairs of mentions whose heads are nouns. IS-IS-A is a boolean feature whose value is true if term(m_1) can be found among the IS-As of term(m_2), false if not, and 'no-isas' if m_2 has no hypernyms. We ignored the DT's sense clusters and used the union of their IS-A sets. ATTR_IS-IS-A works the same way, but the IS-A sets of m_2's attributes are search instead, which were extracted as described in the previous section. To determine the feature SHARED_IS-As we compared each IS-A set of m_1 with each one of m_2, in a similar vein to a WordNet metric by Ponzetto and Strube (2006), and chose the highest similarity score according to their Dice index (Dice 1945) as the feature's value:

$$\max_{i \in I_{m_1}, i' \in I_{m_2}} \frac{2|i \cap i'|}{|i| + |i'|} \tag{4}$$

where I_{m_1} and I_{m_2} are the sets of sets of IS-As for mentions m_1 and m_2, respectively. As with SHARED_PRIOR, we discretized the value by rounding to the first decimal digit.

The set of IS-A patterns used by the JoBimText project includes a copula (*X is a Y*) relation that drastically increases recall, but produces some spurious IS-As like *bit*. The lexicalized features of the Berkeley system are able to assign appropriate weights to those words if they appear as a mention's head. The attribute and overlap features however work with terms that are opaque to the system. We therefore included additional lexicalized variants of both features if they found a match. For ATTR_IS-IS-A, we simply extended the feature with the IS-A. The feature SHARED_IS-As however yields the intersection of two IS-A sets of mentions m_1 and m_2 with the highest similarity according to Equation (4). From this set of shared IS-As, we selected the most specific IS-A as its representative to create the lexicalized version of SHARED_IS-As. The specificity of an IS-A i in a sense cluster $\mathcal{S}$ can be derived from its significance value as recorded by the JobimText DT. This measure was computed by counting how often any term $t \in \mathcal{S}$ was observed co-occuring with i in IS-A patterns in the DT training corpus, and multiplying this frequency by the total number of terms in $\mathcal{S}$ that co-occurred with i (Gliozzo et al. 2013). The higher the value, the more specific the IS-A is assumed to be.

7.3 Context-based expansion

The features we have introduced so far are only concerned with the resolution of bridging mentions, which is why they were restricted to mentions with noun heads. However, we also would like to exploit the knowledge stored in the DT to increase precision in pronoun resolution. Pronouns by themselves carry only a small amount of information, at least when it comes to the English language. Without taking their context into account, we can only determine the number and gender of the entities they refer to. Knowledge about selectional restrictions is required to resolve ambiguous cases.

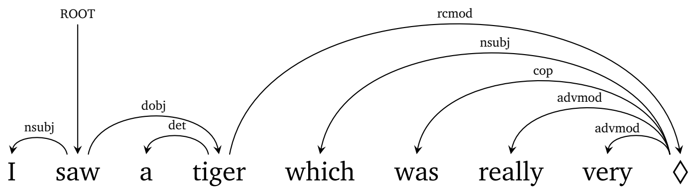
An intuitive approach to modeling selectional restriction using a DT would be to apply context-sensitive re-ranking to the prior expansions of pronouns. However, as they are so prevalent in texts, prior expansions of pronouns are fruitless; they consist mostly of all other pronouns and random noise. The pronoun should rather be treated much like the unknown word ◊ in our toy example from Section 4.1 on Lin's thesaurus, where we guessed the likely attributes of ◊ from the word's context.

We therefore propose an additional type of expansion that targets the surface term's context features, which generate the most likely terms to substitute the surface term. To obtain this expansion, we make use of the fact that the DT's term-context pairs form a language model (Gliozzo et al. 2013). Let C be the set of context features of a term observed in a document and $T = \{t_1, t_2 \ldots, t_n\}$ the set of terms for which there exists a context feature $c_j \in C$ such that the pair (t_i, c_j) is recorded in the DT. We sort the members of T in the descending order of their conditional probability $P(t_i|C)$ and cut off after 200 terms. We call this list the target term's *context-based expansion*, or *C-expansion* for short.

To infer the computation of the probabilities $P(t_i|C)$, first assume a holing system using a context window of size 3 with the hole being in its center. In this case, the set of context features C always contains exactly one element c. We may therefore phrase $P(t_i|C)$ as $P(t_i|c)$, which can be computed like in a regular N-gram model as count$(t, c)/$ count(c) using maximum likelihood estimation (cf. Jurafsky and Martin 2014, p. 91). If we assume the individual contexts of a holing system to be conditionally independent of each other, we can phrase the general case as

$$P(t_i|C) = \prod_{c_j \in C} P(t_i|c_j) = \prod_{c_j \in C} \frac{\text{count}(t_i, c_j)}{\text{count}(c_j)} \tag{5}$$

We will now show that the assumption of context independence has a reasonable foundation in case of the dependency parse holing system used by our main DT.



$P(\text{“fierce”}) = P(\text{saw}|\text{ROOT}) \times P(\text{I}|\text{saw}, \text{nsubj}) \times P(\text{tiger}|\text{saw}, \text{dobj}) \times P(\text{a}|\text{tiger}, \text{det}) \times P(\textbf{fierce}|\text{tiger}, \text{rcmod}) \times$
$P(\text{which}|\textbf{fierce}, \text{nsubj}) \times P(\text{was}|\textbf{fierce}, \text{cop}) \times P(\text{really}|\textbf{fierce}, \text{advmod}) \times P(\text{very}|\textbf{fierce}, \text{advmod})$

Figure 6: Likelihood of the sentence “I saw a tiger which was really very ◊” if the word *fierce* takes the place of ◊, estimated using an order 2 langauge model (slightly modified example adapted from Gubbins and Vlachos 2013).

We adapt the labeled dependency language model by Gubbins and Vlachos (2013), which was created with a sentence completion task in mind. First they assume that a word w_i is conditionally independent of all other nodes and edges in a dependency parse given its ancestor path $A(w_i)$ containing all words and labels from the root down to w_i. The probability $P(w_i|A(w_i))$ can then be calculated as $\text{count}(A(w_i), w_i) / \sum_{w \in V} \text{count}(A(w), w)$, with V being the vocabulary. Because many paths will not occur in the training corpus, they apply the Markov assumption as known from N-gram models to their model and replace $A(w_i)$ by $A^{(N-1)}(w_i)$, which denotes the ancestor path of w_i up to a depth of N. The probability of a dependency parse S containing words S^w is then defined by Gubbins and Vlachos (2013) as

$$P(S) = \prod_{w_i \in S^w} P(w_i|A^{(N-1)}(w_i))$$

If we choose $N := 2$, then we employ what they call an “order 2 model”. The probability of a term taking the place of another one in a sentence can then be determined by calculating the sentence probability after the replacement has been conducted, as shown in Figure 6. Since we are only interested in the probability of a term t_i appearing in the context C considered by the holing operation, we are able to ignore all dependency relations of which the term is not a member, i.e. the factors without a bolded word in Figure 6. By this, we compute the conditional probability $P(t_i|C)$ instead of the probability of the entire sentence S. In an order 2 model dependency language model, $A^{N-1}(w_i)$ consists only of the immediate parent node of w_i and the label that describes their syntactic relationship, which is exactly the information provided by the term-feature pairs of a dependency DT. Thus, Equation (5) above corresponds to an order 2 model.

We made two alterations to the formula: We used the significance values stored for each term-feature pair in the DT instead of frequency counts, and we applied plus-one-smoothing (cf. Jurafsky and Martin 2014, pp. 100-103) to those values to avoid a probability of zero in case a term-context pair is not a member of the DT. This leads to the final equation

$$P(t_i|C) = \prod_{c_j \in C} \frac{\text{sig}(t_i, c_j) + 1}{\sum \text{sig}(*, c_j) + V}$$

with V being the number of distinct context features in the DT. In practice however, the nominator, which is the same for all terms can be omitted, as the values are only required to be proportionally correct for ranking the expansion terms.

For an example C-expansion produced by the dependency DT, consider the sentence:

(11) The children like drinking $\Diamond$.

The placeholder's expansion has *beer*, *mix*, *wine*, *it*, and *idea* as its top five words. The example shows the shortcoming of a dependency language model of order 2. Only the underlined words are in a relation with the placeholder and therefore consulted by the algorithm, which is why we obtain beverages not suitable for children. Most often, the other arguments of a verb impose additional selectional restrictions (Calvo et al. 2009; Lenci 2011). If we had e.g. captured the context in a window instead of exctracting dependency relations, we would likely have gotten better results in this particular case. On the other hand, if the context window is too small, the more informative words of a sentence may be missed, which was the reason why Gubbins and Vlachos (2013) created their dependency language model in the first place. For example, even a context window with four words to the left of its hole would fail to produce any useful substitutes for the placeholder in the sentence depicted in Figure 6.

For the ensuing feature IN-C-EXPANSION we used the discretized rank of term(m_1) in the C-expansion of context(m_2). If m_1 was a pronoun mention, we C-expanded context(m_1) as well and used the highest-ranked content word as the query term instead. This way, pronouns with identical lexemes can be compared with respect to their semantic compatibility, a feature that would be impossible to model with non-generative vector spaces. We also applied the feature if both mentions' heads were nouns, even though one might argue that their prior expansions render a C-expansion superfluous. However, a C-expansion allows to take a noun's contextual role (Bean and Riloff 2004; cf. Section 5.2) into account, and becomes vital information if at least one head term is not a member of the DT's vocabulary, which is often the case with proper names.

Note that the feature rather models selectional *preference* than selectional *restriction*, as the smoothing allows terms to be members of the C-expansion even if they are semantically compatible only with some of the context features, but not all of them. We applied a cut-off after 200 terms for the same reason.

It is rather unlikely that the exact term of m_1 appears in m_2's small C-expansion, even if the context is appropriate. Bergsma et al. (2008a) used automatically obtained word clusters to map verb arguments to their semantic classes for smoothing. Since the prior expansion of a word acts like its (sense-unaware) cluster, we assume the C-expansion to be constituted of representatives of those clusters. Thus, we experimented with checking whether term(m_1) is a member of the semantic clusters preferred by the context. We prior expanded all content words in the C-expansion of m_2 and searched those expansions for term(m_1). We tried out both a boolean feature that fired if the query term was found among any of those expansions, and a feature that uses the explicit rank calculated as the sum of C-expansion rank and transitive prior rank, but noticed no improvement on the coreference metrics. We therefore discarded the transitive lookup feature entirely.

8 Evaluation

The evaluation of a coreference resolution system is not as straightforward as it is in the case for other annotation tasks like POS labeling. A simple solution would be to require the system to add a label to each mention identifying the real-world entity it refers to, but this is impossible, as the entities referenced in a document are latent. Rather, the clusters produced by the system form the entities and have to be compared with the gold annotation.

This leads to a wide range of possible errors the system could have made. For example, the system might consider mentions as coreferent which are non-anaphoric, resulting in a spurious entity. A different error occurs when a set of coreferent mentions is correctly identified as anaphoric, but the system assigns them to two different entities. Coreference resolution is often understood as an intermediate task,[20] and if the system is embedded into another one, the evaluation procedure applying to the enclosing task would be used to compare competitors, thereby indirectly weighting the individual error types. However, since a measure was needed that allows a comparison of coreference systems without a higher task, various *coreference evaluation metrics* have been proposed, which base their calculations on different elements (links, mentions and entities) and vary in the weight they assign to different error types.

The rest of this section is organized as follows: We introduce the three coreference metrics used to evaluate the systems: MUC, B^3 and $CEAF_e$, in Section 8.1. We show our results for various features on the development and test set of the CoNLL-2011 shared task in comparison to Berkeley's Final model in Section 8.2. We also investigate into the negative impact of context-sensitive re-ranking on the system's performance in this section.

In Section 8.3, we report the results of a different evaluation scheme, pairwise linkage, that is only possible for a system that selects latent antecedents. Furthermore, we take a look at our system's performance on bridging mentions.

In Section 8.4, we perform a manual analysis on a small subset of errors to identify the system's major problems. We automatically classify and count different types of errors made by our system and compare them to the baseline using a tool by Kummerfeld and Klein (2013) in Section 8.5. As precision is our greatest problem, we experiment with a filter to remove likely non-coreferent terms from the DT's prior expansion in Section 8.6. Finally, Section 8.7 compares the dependency-based holing operation with an approach using a context window.

8.1 Coreference evaluation metrics

We chose the three metrics that were used during the CoNLL-2011 shared task (Pradhan et al. 2011) to evaluate the system: MUC (Vilain et al. 1995), B^3 (Bagga and Baldwin 1998) and $CEAF_e$ (Luo 2005). The authors of each scoring algorithm claimed to have fixed errors made by their predecessors, so one might expect the most recent of the three, $CEAF_e$, to be the most suitable. However, an informal survey conducted by the CoNLL-2011 shared task organizers revealed that neither metric is preferred in favor of the other. Each metric focuses on another aspect of coreference, and the selection depends on the type of task in which coreference is needed. Therefore, they took the average of the three metrics' F_1 values to rank the participating systems (cf. Pradhan et al. 2011). We adopt this approach when we present our results in the next section.

All three metrics have in common that they compare a set of key entities $\mathcal{K}$, which represent the test set's gold annotation, with the set of response entities $\mathcal{R}$ from the system's output. Furthermore, they all report precision and recall in percentages such that a perfect system achieves 100%. The meaning of precision and recall differ from scheme to scheme, depending on the structure it focuses on: links, mentions, or entities. A score combining precision and recall can be obtained by calculating the F_1-measure[21] as their harmonic mean (Shaw Jr et al. 1997).

Like Pradhan et al. (2014), we will refer with K_i to the i$^{\text{th}}$ entity in the key set $\mathcal{K}$ and with R_i to the i$^{\text{th}}$ entity in the response set $\mathcal{R}$.

The **MUC** metric by Vilain et al. (1995) was initially designed for the evaluation of systems on the corpus of the same name. It compares the *links* present in the coreference chains generated by the key and response entities. A link exists between each pair of neighbors in the chain, thus the number of links in a chain resp. entity containing n elements is $n-1$. Recall is defined as the ratio between the number of correct links in $\mathcal{R}$ and the total number of links in $\mathcal{K}$, and precision is the ratio between correct links in $\mathcal{R}$ and the total number of links in $\mathcal{R}$. A link is said to be correct if the mentions it connects are coreferent. The number of correct links is determined by calculating the number of missing links. For this, key and response are partitioned in relation to each other. Let $p(S)$ be a function that partitions a single key entity S by intersecting it with each entity in the response, keeping only non-empty sets, and assigning each remaining mention that was not contained in the response to a singleton set. This partition

[20] As Appelt (1999) puts it: "Nobody is interested in coreference for its own sake".

[21] $2PR/(P+R)$ with P = precision and R = recall

indicates how many links have to be added to the response entity such that it contains all links from the key entity. Each link corresponds to a merge operation on two elements in the partition. As there are $|p(S)|$ elements in the partition, $|p(S)|-1$ merge operations have to be performed until all sets in the partition are united. To compute precision, the roles of key and response entity are reversed. Let $p'(S')$ be a function that partitions a single *response* entity S' with respect to the set of key entities in a similar vein to $p(\cdot)$. Recall and precision for all entities can then be computed as follows (Vilain et al. 1995; Pradhan et al. 2014):

$$R_{\text{MUC}} = \frac{\sum_{i=1}^{|\mathcal{K}|}(|K_i| - |p(K_i)|)}{\sum_{i=1}^{|\mathcal{K}|}(|K_i|-1)} \qquad P_{\text{MUC}} = \frac{\sum_{i=1}^{|\mathcal{R}|}(|R_i| - |p'(R_i)|)}{\sum_{i=1}^{|\mathcal{R}|}(|R_i|-1)}$$

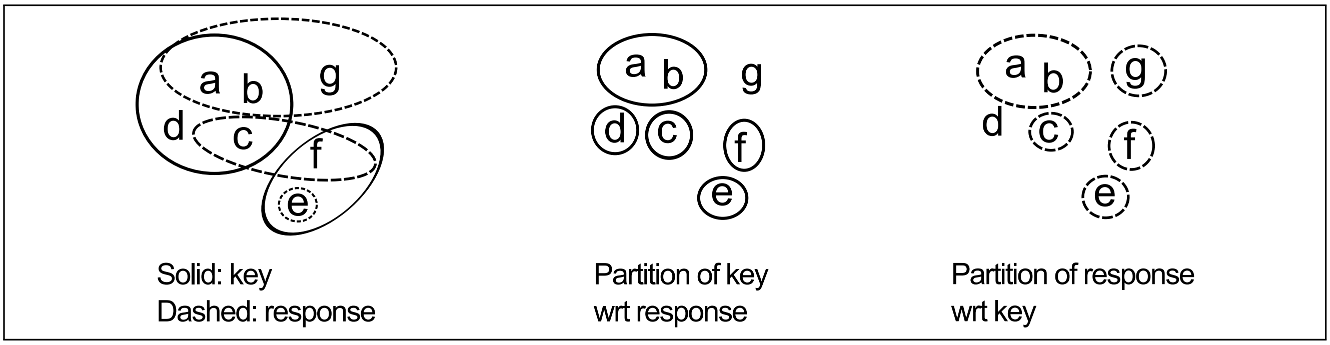


Figure 7: Partitioning entities to compute the MUC score (based on Pradhan et al. 2014). Left: initial situation with key and response entities. Center: application of $p(S)$. Right: application of $p'(S')$.

For an example of the partition process and the computation of recall and precision, we use the entities depicted in Figure 7. First, consider the key entity containing the mentions a, b, c and d with size 4. This means that 3 links, namely $\{a-b, b-c, c-d\}$, are required to establish the entity. Its partition with relation to the response is $\{\{a,b\},\{c\},\{d\}\}$, as can be seen from the figure in the middle. 2 merge operations are required to unite all sets in the partition. Thus, $3-2=1$ links were correct in the response entity, and its recall is 1/3. Likewise, to determine the precision of the response entity $\{a,b,g\}$ containing 2 links, we partition it with relation to all key entities, resulting in the partition $\{\{a,b\},\{g\}\}$ seen in the rightmost image. One merge is required to reconstruct the original entity, thus its precision is $(2-1)/2 = \frac{1}{2}$.

Bagga and Baldwin (1998) critized the MUC metric for producing equal precision scores for a system that falsely conjoins two large entities, and a system that wrongly conjoins a large and a small one. They argue that the first error is more severe and proposed the **B^3** measure to correct this issue. B^3, in contrast to the link-based MUC metric, operates on mentions and reduces precision proportional to the number of non-coreferent mentions in the predicted entity containing it.

They reuse the partition functions from the MUC metric to explain their formal definition, however Pradhan et al. (2011) provide us with an intuitive informal one that leads to a more concise equation: The B^3 recall for a single mention m is the number of correct mentions in the response entity containing m divided by the size of its (true) key entity, and precision is the number of correct mentions in the response entity containing m divided by its own size. Recall and precision for complete key and response sets is computed as the average of the individual mention scores. We reproduce the equations for B^3 document recall and precision from Pradhan et al. (2014, variable names adjusted):

$$R^3_{\text{B}} = \frac{\sum_{i=1}^{|\mathcal{K}|}\sum_{j=1}^{|\mathcal{R}|}\frac{|K_i \cap R_j|^2}{|K_i|}}{\sum_{i=1}^{|\mathcal{K}|}|K_i|} \qquad P^3_{\text{B}} = \frac{\sum_{i=1}^{|\mathcal{K}|}\sum_{j=1}^{|\mathcal{R}|}\frac{|K_i \cap R_j|^2}{|R_j|}}{\sum_{i=1}^{|\mathcal{R}|}|R_j|}$$

While B^3 fixed the shortcomings of the MUC algorithm, it introduced a new problem, as pointed out by Luo (2005): As a result of the iteration over mentions, the same entity might be evaluated multiple times, which leads to a distortion of the results. He therefore presented with **CEAF** a slightly more sophisticated, entity-centric metric. It requires as a first step that the key and response entities are in pairwise alignment such that each key entity has at most one response entity as its partner and vice versa. The second constraint is that the pairs are aligned in a way that maximizes their total similarity score, which is defined as the sum of their individual similarity value according to some measure $\phi(R_i, K_j)$ comparing a response entity R_i to a key entity K_j. The best pairings can be found with the help of a maximum bipartite matching algorithm.

Luo (2005) proposed various similarity measures. We apply the entity-centric variant of CEAF, called CEAF$_e$, which was used by the official CoNLL-2011 shared task scoring scheme (Pradhan et al. 2011). The similarity ϕ_e between a response entity R_i and a key entity K_j is computed as

$$\phi_e(R_i, K_j) = \frac{2|R_i \cap K_j|}{|R_i| + |K_j|}$$

Let g^* contain the alignments of the best pairing obtained as described above. Recall and precision are then computed by dividing the summed similarity scores in g^* by the number of key or system entities, respectively (Luo 2005):

$$R_{\text{CEAF}_e} = \frac{\sum_{(r,k)\in g^*} \phi_e(r,k)}{|\mathcal{K}|} \qquad P_{\text{CEAF}_e} = \frac{\sum_{(r,k)\in g^*} \phi_e(r,k)}{|\mathcal{R}|}$$

8.2 Metric results

To obtain MUC, B^3 and CEAF$_e$ metric scores for our system output we used the coreference scorer by Pradhan et al. (2014)[22] in version 7, which represents the reference implementation of those metrics. Differences in values reported here and in Durrett and Klein (2013) are due to changes made to the system since its first release and the fact that previous versions of the scorer were faulty. Table 1 shows the results of the following systems, which were trained on the training set and tested on the development set:

Baseline Berkeley's Final system without any further modifications.

Prior Adds the features PRIOR and SHARED_PRIOR to the baseline.

Prior+Isa Extends Prior by adding the features IS-IS-A and SHARED_IS-As.

Prior+Isa+Ctx Adds the feature IN-C-EXPANSION to the system above.

Best Set Contains all of the above, as well as the attribute-centric feature ATTR_PRIOR and ATTR_IS-IS-A.

Dummy Contains all the features from Best Set, but accesses a mock DT implementation. The holing operation is performed as usual, but prior expansions contain only the expanded term itself. The C-expansion of a mention and its set of IS-A clusters are always empty.

The significance of improvements, especially if they are as small as ours, has to be established. After all, the score differences could be simply due to chance, which means there would actually be no improvement at all. A *significance test* allows to compare two systems A and B. It ascertains a probability p for how likely we falsely reject the hypothesis that system B is worse than system A. If p is below a threshold (commonly 0.05), we can say that system A is signifcantly better than system B.

We employed pairwise bootstrap resampling (Koehn 2004) which works as follows: A new test set of equal size is created by randomly drawing documents from the original test set with replacement.[23] The metric scores for both systems are calculated on this new set and it is noted whether system A outperformed system B for each score separately. Both steps, resampling and evaluation, are repeated N times, with N being a large number (usually 1000). If system A was better on at least $(1-p)$% of the test sets, one can say with confidence $(1-p)$% that system A outperformed system B in terms of the measure in question.

We wanted to validate whether distributional semantics were responsible for the system's improvement, or just the holing operation in combination with our feature layout. We therefore trained and tested our Best Set system containing all successfull features while redirecting all DT queries to a mock implementation. Hence, this Dummy

[22] http://code.google.com/p/reference-coreference-scorers

[23] Note that with "test set" we refer to the set the model is tested on, which in our case is either the corpus' development or test set.

	MUC			B^3			CEAF_e			
	P	*R*	F_1	*P*	*R*	F_1	*P*	*R*	F_1	*Average*
Baseline	68.59	63.41	65.90	60.80	53.51	56.92	57.05	55.42	56.23	59.68
Dummy	68.88	63.60	66.13	61.24	53.53	57.12	57.30	55.63	56.45	59.90
Prior	68.52	63.81	66.08	60.75	53.88	57.11	57.25	55.89	56.56	59.92
Prior+Isa	68.74	64.21	66.40	60.85	54.48	57.49	57.53	55.91	56.71	60.20
Prior+Isa+Ctx	68.77	64.33	66.48	60.57	54.79	57.53	57.52	55.86	56.68	60.23
Best Set	**69.27**	**64.46**	**66.77**	61.28	**54.80**	**57.86**	**58.25**	**56.09**	**57.15**	**60.59**

Table 1: Metric scores of the Berkeley system's Final model (the baseline), compared to models using various combinations of the features described in Chapter 7, when tested on the development set. Highest values in each column are bolded if they are significantly better than those of the Baseline system according to a paired bootstrap resampling test with $N = 10\,000$ and $p = 0.05$.

system did not make use of any distributional knowledge. It is able to perform as well as the Prior system whithout sharing with it the loss of precision, primarily because of the attribute-centric features. With a dummy thesaurus, they are equivalent to checking whether the head term of mention A is among any of the attributes of the head of mention B. For example, the system is able to link *Barack Obama, president of the United States* with *the president* by considering the appositive relation present in the first mention. Although the Berkeley system checks whether a mention's head is contained in another's entire span, attribute features are more precise, as they take only the dependency relations of the head into account. Yet, although a whole 27% of the allegedly bridging mentions can be resolved just by looking at other words belonging to the mention NP than the head (cf. Figure 9 on p. 34), the average increase in recall when compared to the baseline is smaller than the Prior system's, and none of these increases was significant.

It is notable that the Prior system's precision, which features only the head expansion without any reranking, decreased in comparison to the baseline. However, this is consistent with observations made by Versley (2007) and was expected from a noisy knowledge source. The CEAF_e precision score seems to have gotten better, but the increase is not signifcant. The loss in precision for the other two metrics, -0.07 and -0.05 for MUC and B^3 respectively, is small in comparison to an average gain in recall of 0.41 percentage points.

Unsurprisingly, adding IS-A features to the Prior system leads to another boost in performance. Thanks to their high quality, the system could improve its recall over Prior on MUC and B^3 metrics, but not on $CEAF_e$. Our system is prone to unite two unrelated entities, which is punished especially by the B^3 precision evaluation as intended by design (Bagga and Baldwin 1998), but also affects the CEAF_e recall computation. As CEAF_e allows only one-to-one mappings between key and response entities, the falsely conjoined entity will be paired with the largest key entity contained, leaving the smaller one uncredited. Whether the risk of merging unrelated entities is acceptable depends on the application the resolution system is embedded in. For example, if we want to create a facts database, precision is of the essence: We rather like to learn learn fewer facts than include false knowledge. In that case our features are detrimental to the host system's performance.

As can be seen by the partially decreasing scores of the system Prior+Isa+Ctx, the context-based expansion feature appears not to be sufficient to model selectional preference. We noticed in our manual analysis (Section 8.4) that the feature makes the system link non-coreferent pronouns.

Finally, the addition of attribute features, forming the Best Set system, raises the scores approximately by the same margin as the Dummy system in comparison to the baseline. However, for the first time, all scores with the exception of B^3 precision increased significantly compared to the baseline. We assume that the DT and IS-A features provide the system with semi-redundant information regarding mentions with attributes. After all, not every attribute of a mention can be considered its possible substitute. For example, we cannot link *Bush* to *Bush administration*, as *Bush* only acts as a modifier in this case by specifiying the administration in greater detail. On the other hand, the coreference of *actor Humphrey Bogart* and *the actor* can be established more easily by the system, as it has evidence from both the attribute and the corresponding IS-A pair.

So far, we did not make use of context-sensitive re-ranking. Biemann and Riedl (2013b) showed that this feature is able to improve precision significantly in a lexical substitution task. Therefore, we expected it to improve our coreference expansion as well, since coreferent mentions are mutually substitutable (Mitkov 2002, p. 7), thereby mimicking the task. Table 2 lists the change in MUC, B^3, CEAF_e, and average F_1 scores after adding re-ranking to

	MUC	B^3	$CEAF_e$	Avg
Best Set -PRIOR,+RERANK	-.14	-.13	-.15	-.14
Best Set +RERANK	-.06	-.01	-.02	-.03

Table 2: Change in F_1 metric scores after modifying the Best Set system to include the context-sensitive re-ranking feature.

the Best Set system. In the first line, the PRIOR feature was replaced entirely by RERANK, while in the second line RERANK was added side-by-side. Both modifications showed no improvements on the scores.

To find out why, we collected the ranks of each mention head from the training set in both expansions of all its preceding mentions in the training set. Only pairs of mentions with noun heads were considered. We accumulated the frequency of ranks over all mentions separated into expansion type and whether the mention and its predecessor were coreferent or not. The factor by which the re-ranking changes the rank frequencies of a prior expansion is displayed in Figure 8.

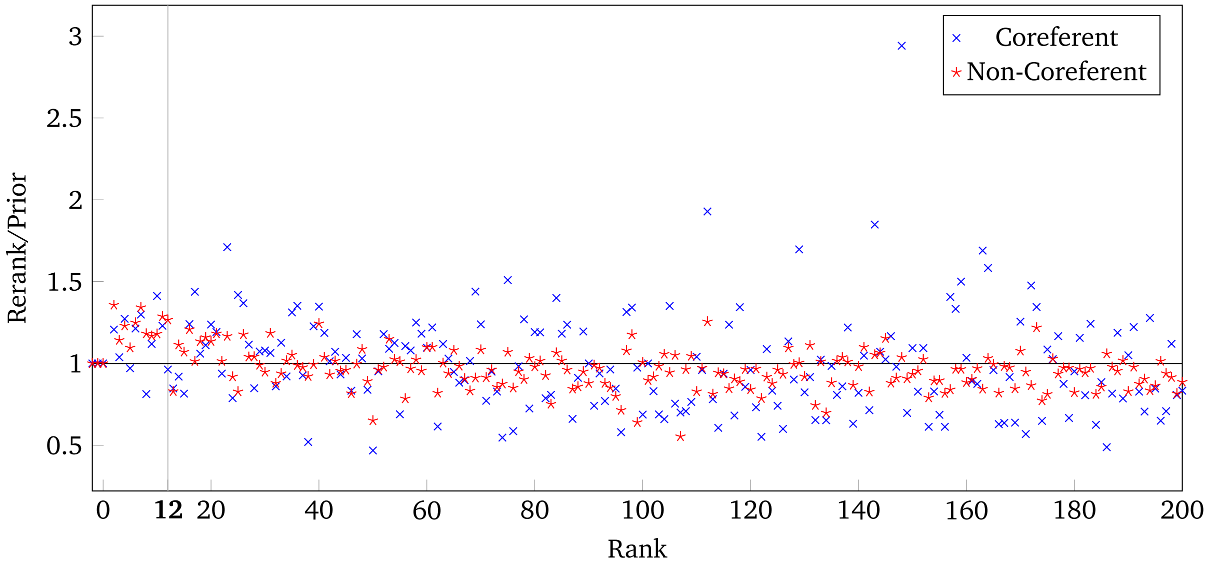


Figure 8: Factor of change in rank frequencies after applying context-sensitive re-ranking to prior expansions. Values below 1 indicate that the number of mentions found in the corresponding rank decreased in comparison the the prior expansion.

We observe that context-sensitive re-ranking had not the desired effect on coreferent mentions (blue crosses). If the feature had worked as planned, these mentions should have moved to the expansion's top, i.e. the frequency of occurrence of those ranks should have risen, while ranks on the right half of the plot should have been observed less often. In reality, the direction of frequency changes fluctuates over the full spectrum. While we can see a cluster at the top 60 for which the frequency of coreferent mentions has mostly increased, simultaneously there are also a lot lower ranks with more mention pairs than before. With respect to non-coreferent mentions, the frequency changes are less extreme, presumably because this sample was roughly a hundreth times larger and therefore less prone to outliers. Unfortunately, this class appears to be the one that profited from re-ranking: the bottom ranks are less frequent than they were in the case of prior expansions, while the higher ranks now contain many more mentions. A particular nuisance are the top 12 ranks, for which the number of non-corefent mention pairs grew without exception. Being faced with this increased noise, the system is not able to assign higher weights to top ranks without risking a loss in precision. Thus, the RERANK feature is useless in its current form.

We assume the reasons for this to be twofold. First of all, except for ties, a term's semantic similarity value is ignored by the re-ranking operation. This may cause a rare, non-coreferent term at position 200 of a prior expansion to become first if it fits the context well. While in theory the weights learned by the classifier for the two features PRIOR and RERANK, when used in parallel, should counter this, a lot of valuable information gets lost by our binning and using ranks instead of raw values. The second reason is that the increased frequencies of lower ranks for coreferent mentions are likely caused by terms which are contextual incompatible because the context is too specific. Consider the following example:

(12) *Last year we visited [West Virginia]. [This state] has many landmarks.*

If we apply context-sensitive re-ranking to the first mention's head *Virginia*, the algorithm will factor in the noun modifier *West* as a context feature. As a consequence, the term *state*, which is already far down at position 44 in the prior expansion, will be ranked even lower in favor of words like *Alaska* or *Arizona*. Our misconception here was to allow words from the mention itself to be used as context features. This contradicts the notion of a coreference substitution test, in which *mentions* are substituted for each other, not single head words.

To address these problems, we made two changes to the re-ranking scheme. First, we altered the re-ranking formula to combine both the context score (Eq.(3), p. 15) and similarity score by taking the geometric mean of both values. To obtain this similarity score, we simply use the number of shared contexts between two terms as provided by the DT's prior expansion. Second, we removed from the set of context features considered for re-ranking those which contain terms from the mention. Unfortunately, this new scheme led to even worse results; the average score decreased by 0.23 compared to Best Set.

Therefore, we report our final results on the test set in Table 3 using the Best Set system without context-sensitive re-ranking.

	MUC			**B^3**			**CEAF$_e$**			
	P	*R*	*F$_1$*	*P*	*R*	*F$_1$*	*P*	*R*	*F$_1$*	*Average*
Baseline	**69.34**	65.69	67.46	**58.38**	52.66	55.37	53.33	53.48	53.40	58.75
Best Set	68.78	**67.00**	**67.88**	57.34	**55.05**	**56.17**	**54.28**	**54.53**	**54.41**	**59.49**

Table 3: Final results obtained after training both systems on both the CoNLL-2011 shared task training and development sets and evaluating them on the test set. Bolded values indicate that the system achieved a significantly better score than the other according to a paired bootstrap resampling test with $N = 10\,000$ and $p = 0.05$.

Our system is more imprecise than the baseline, but this is balanced by an increased number of recalled links, mentions and entities, which results in an overall better F_1 score across all metrics. We achieved an improvement of +0.42 on MUC, +0.80 on B^3 and +1.01 on CEAF$_e$ F_1. That precision went up on the CEAF$_e$ metric demonstrates why one should not rely on a single coreference scoring scheme: The CEAF$_e$ precision formula underestimates spurious system entities. If the system forms entities from mentions not present in the key set, precision will decrease proportional to the total number of those new entities, regardless of their size. Even a system that creates one big entity from all words not annotated in the key would be penalized only slightly by CEAF$_e$. The other two metrics pay better attention to the fact that our system is prone to create such extra entities (see Section 8.5).

In comparison, both the pattern-based approach by Haghighi and Klein (2009) and the one using the top five words in a DT by Ng (2007) led to no significant loss in precision. Haghighi and Klein (2009) was able to increase recall by 7.4 on average, a much larger margin than what we achieved. However, their and our results are not entirely comparable, since they were obtained on different datasets (ACE and MUC corpora).

8.3 Pairwise linkage and bridging mentions

Metrics are useful if one wants to rank competing systems or quickly evaluate features during development. For a detailed error analysis however we need to know how well the system handles different kinds of mentions. To keep things simple, we treat the resolution system as a binary classifier. We consider a mention of a category of interest either correctly resolved or not. However, there is no notion of "correctly resolved" for a mention in the task of coreference resolution. A strict definition would deem a mention m only correctly resolved if it is coreferent with all its antecedents in the system coreference chain accoding to the gold annotation. Clearly, this evaluation scheme is too harsh, as even state-of-the-art systems are far from returning such perfect entities.[24] A relaxed definition selects a single antecedent and checks whether both mentions were in the gold entity. This type of evaluation is facilitated by the Berkeley system, as it predicts latent antecedents.

[24] The large number of divided entities, conflated entities, extra mentions and missing mentions depicted in Figure 10 below bear witness to this fact.

	Berkeley	Dummy	Prior	Best Set
Nominal, anaphoric	65.54%	65.67%	66.43%	**66.49%**
Nominal, no antecedent	94.80%	**94.85%**	94.64%	94.79%
Pronoun, anaphoric	75.84%	75.93%	76.07%	**76.60%**
Pronoun, no antecedent	61.59%	**61.75%**	61.70%	60.60%

Table 4: Performance of different systems evaluated on latent antecedent links in the development set.

Table 4 shows in how many cases the correct antecedent was selected by the systems under discussion for two types of mentions: nominals and pronouns. Unsuprisingly, the features based on distributional semantics help to increase the number of correctly resolved anaphoric nominals, but introduce spurious links. Our single feature targeting selectional preference, IN-C-EXPANSION, was actually detrimental in the category it was designed for. While the Best Set system was able to slightly increase the number of correctly resolved anaphoric pronoun mentions compared to the baseline, it considers far more pleonastic and cataphoric pronouns as coreferent with previous mentions. An elimination test showed that the feature is nonetheless useful, presumably because it is the only available distributional feature for mention heads which neither are part of the DT's vocabulary nor possess any attributes.

Our main focus was on bridging mentions. Since only 7.6% of all mentions in the development set are bridging, the impact of their correct resolution on the absolute metric scores is minor. We therefore present precision and recall for this category alone, and also take a look at the different types of bridging resolved. For this, we manually labeled all bridging mentions in the development set either with the semantic relation it holds with its last nominal antecedent, or with a category that represents the information from the mention itself or its context that should primarily be used to resolve it. For example, we annotated the bridging mention *[The pet]* with the hypernymy label in the coreference chain *The dog – it – it – The pet*. We used the Berkeley system's head prediction to detect bridging mentions in the development set. We considered 12 of those mentions non-bridging, as their heads were pronouns or demonstratives incorrectly labeled as nouns by the POS system. The remaining 1071 were each annotated with one of the following labels:

ACRO One head is an acronym of the other, like in the case of *PRC* and *People's Republic of China*.

ATTR One mention already contains the other mention's head, e.g. *the International Red Cross Organization* and *the Red Cross*. Frequently observed in case of full names referenced by given names.

CANBE The head of one mention is a possible property of the other mention's head, often likely, but not inherent in its meaning. For example, an *agreement* can be a *concession* for one party, but the words are not in a hypernymic relation.

DATE Bridges like *today – the 30th* which could be resolved with the help of an NER system that labels date and time entities.

DISC Resolution of the bridging mention requires discourse knowledge from the document, e.g. *my mother* and *Thelma Wahl*. This category also includes ornate paraphrases, for example calling *Hong Kong* the *Hollywood of the East*.

HEAD Both mention heads are identical, but the system selected the wrong word as the head for at least one mention. This category was frequently observed for transliterations of asian names, in which the family name preceded the given name.

HYP One mention head is a hyponym or an instance of the other.

HYPATTR A hyponym or hypernym of one of the mention's heads is contained in the other mention's NP, e.g. *Doctor Hunter* and *the physician*.

LEMMA Both heads have the same lemma. This class is difficult to resolve as a disagreement in number between two mentions is normally negative evidence for coreference.

MET The heads are in a metonymous relationship, e.g. *the Japanese government* and *Japan*.

SYN The two mention heads are synonyms or near-identical, e.g. *dad* and *father*. Also included are spelling variants of the same name.

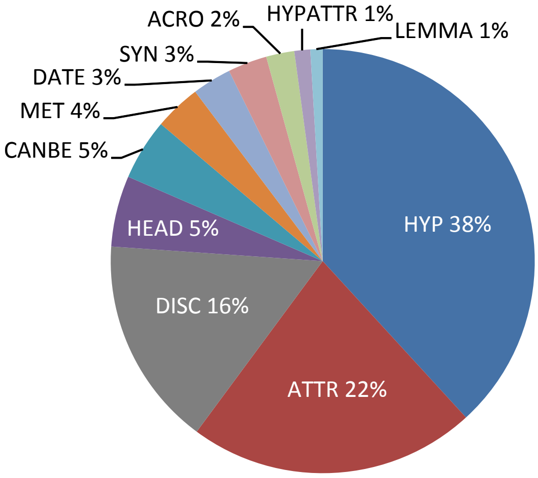


Figure 9: Distribution of bridging types occurring in the development set.

Figure 9 shows the distribution of the various bridging types in the CoNLL-2011 development set. Note that hypernyms (HYP) and words in a mention different from its head (ATTR, HYPATTR) together represent 61% of the bridging relations. They were targeted by dedicated features in our system, those based on IS-As and those extracting attributes, and primarily responsible for the system's increased recall (see below). Unfortunately, discourse bridges, which require either an understanding of the text or extensive background knowledge to be resolved, also make up a large portion (16%). Consider the following example:[25]

> *George W. Bush has met with* ***Al Gore*** *in Washington. The two men met for just 15 minutes at the Vice President's official residence. It's the first time they've been face to face since the bitterly contested presidential election was finally concluded last week. Bush went into the talks with* ***his defeated rival*** *after meeting with President Clinton earlier today.*

Neither system was able to correctly link the coreferent mentions here marked in bold. To resolve them, the systems would have to know that both Bush and Gore were presidential candidates in the same election and that this fact makes them rivals.

It is interesting to note that transitive attributes (HYPATTR), which originally motivated our attribute features, are in fact the second-smallest class observed in the development set (the smallest one being LEMMA).

We present the number of bridging mentions correctly resolved by the baseline system and Best Set, broken down by bridging type, in Table 5. We consider a bridge resolved if the mention and its last gold antecedent with a noun head were labeled as coreferent by the system. For example, the bridge *The dog – it – it – The pet* is correctly resolved if the system assigned *[The dog]* and *[The pet]* to the same entity.

	ACRO	ATTR	CANBE	DATE	DISC	HEAD	HYP	HYPATTR	LEMMA	MET	SYN	$\sum$
Baseline	2	89	0	1	8	28	54	0	0	7	5	194
Best Set	3	110	5	2	7	29	121	2	1	10	8	298
Total	23	235	50	32	171	58	409	13	10	38	32	1071

Table 5: Bridging mentions resolved per type, compared to their total number in the CoNLL-2011 development set.

In total, we resolved 104 bridging mentions more than the baseline. That the baseline system resolved any bridging mention at all may seem unexpected, as it does not make use of any semantic information. However, the coreference chain of two bridging nominals may have pronouns between them, which can act as bridges by themselves. Furthermore, the Berkeley system is able to exploit syntactic clues, frequently observed lexical pairs,

[25] Taken from document "pri_0090" from the CoNLL-2011 development set.

and the distance between mentions. Quite often, the assumption that the previous mention is an antecedent is a strong baseline (cf. Versley 2007).

We observe better recall among all types of bridging mentions except those that require discourse knowledge to be resolved. The greatest strength of our system is the detection of hypernymic links. More than twice as many bridges from this category were resolved compared to the baseline. We found that IS-A patterns were not solely responsible for this increase. For example, the pairs of coreferent mentions *(a marketing study, the survey)*, *(the balloting, elections)* and *(the insurrection, the Oct. 3 failed coup in Panama)* could only be detected thanks to the DT's prior expansion.

Contrary to our expectations, the number of CAN-BE relations detected did not increase that much, despite them being captured by IS-A patterns as well. Examples from this category include *the defendants – the men* and *his testimony – his assertions*. The mentions from the former pair have in common that they are persons, and the latter two mentions are abstract concepts. This suggests that the system could profit from a distributional approach to semantic class induction in a similar vein to Ng (2007). He ran a named entity recognizer on a corpus and propagated the label of NPs to their appositives. In our case, a classifier could learn the correlation between clusters from the DT and semantic classes in an annotated corpus. We didn't look further into this approach since we did not want to introduce more labeled training text in addition to the coreference corpus.

In addition and in reference to Versley (2007), we report precision, recall and F_1 score for bridging mentions separately in Table 6. Recall is the resolved bridging mentions' proportion of the total number of bridges, and precision is calculated by dividing the number of spurious bridging links by the total number of bridging links created by the system.[26]

	P	R	F_1
Baseline-Dev	32.66	18.10	23.29
Best Set-Dev	**39.37**	**27.70**	**32.52**
Baseline-Test	36.38	16.07	22.23
Best Set-Test	**38.00**	**26.08**	**30.94**

Table 6: Precision, recall and F_1 scores on bridging mentions in the CoNLL-2011 development and test sets, achieved by the baseline and our best system.

While we were able to increase the number of resolved bridging mentions, recall is still far from an acceptable value. As Durrett and Klein (2013) pointed out, the Berkeley system will not give much weight to semantic features if they don't provide strong evidence of coreference. Many non-coreferent mentions with semantically similar heads in the CoNLL corpus made the system have only low confidence in our distributional features. That our system also shows better precision than the baseline may be surprising at first. After all, we saw above that precision went down for all DT-based systems. However, we calculated precision in Table 6 only on bridging *links*, yet a single wrong bridge may create many more spurious links as a consequence. For example, the coreference chain *the house – the man – the man* contains only one bridging link, the one between the mentions *[the house]* and *[the man]*. The second *[the man]* is not bridging by our definition, as it shares an identical head with its predecessor. Also, because of the way how precision is defined in general, a higher value does not automatically imply that our system made fewer errors than the baseline. In fact, it made *more* (466 compared to 404 on the development set), but the increase of recalled mentions led to a relatively "more precise" response.

8.4 Manual error classification

A superior solution to a problem is one that contains less errors. Improving an existing system can thus be formulated as reducing the number of errors it makes while avoiding the introduction of new ones. By categorizing errors into different types, it is possible to identify false leads confusing the system or missing knowledge required to correctly link certain mentions.

We therefore performed a manual classification of errors made by the Best Set system on the development set. In the scope of this analysis, an error is said to occur if a mention and its last antecedent in the system coreference chain are not coreferent in the key set, or if a mention is erroneously classified as non-coreferent when in fact it is. We randomly selected 100 of those incorrect mentions, and additonally 100 errors made by our system but resolved

[26] This time, we included the incorrectly labeled pronouns and demonstratives into the calculations.

correctly by the baseline. This distinction allows to estimate which errors are common to both models, and which were a result of the addition of our semantic features. We examined the features and their weights assigned by the system to the false and correct links to find out what (presumably) caused the system to make the wrong decision. The errors were then classified into categories corresponding to the assumed error source.

Error type	Frequency
Non-coreferring	13%
Discourse	12%
Gold error	12%
Parsing	11%
Syntactic clues	10%
Semantics: recall	8%
Semantics: precision	7%
Missing confidence	6%
Clashing attributes	3%
Salience	3%
Mention nesting	2%
Other/Unknown	13%

(a) 100 random errors

Error type	Frequency
Semantics: precision	23%
Parsing	11%
Mention nesting	9%
Non-coreferring	9%
Discourse	7%
Missing confidence	6%
Semantics: recall	5%
Clashing attributes	4%
Gold error	4%
Salience	4%
Weights	4%
Syntactic clues	3%
Other/Unknown	11%

(b) 100 random errors not made by the baseline

Table 7: Distribution of error classes observed in a manual analysis, ordered by their frequency.

Tables 7a and 7b show how frequently various types of errors appeared in the manual analysis. Rare classes were subsumed under the "Other" category. These include cases like guessing the gender of a mention wrong or transitive errors which require an entity-mention model to be addressed properly. In the remaining paragraphs of this section, we describe the error types and name potential approaches to avoid them. The numbers in parentheses repeat the frequency of that class in the general error set (first number) resp. the one containing errors not made by the baseline (second number) from Table 7.

Detection of non-coreferring mentions (13/9). The system has problems to detect mentions that do not refer to a previous mention. It has a disposition to reject the hypothesis that a mention starts a new cluster if a mention has a pronominal head or its head is preceeded by a definite determiner (e.g. *the*). We observed three different kinds of non-referring mentions:

Non-coreferring pronouns: This includes the pleonastic it like in *it rains*, which the system incorrectly classified as anaphoric in 54% of its occurrences, and the generic you like in *you reap what you sow*. The detection of non-referential pronouns is a well-researched topic. Bergsma et al. (2008b) proposed a distributional approach, which we tried to imitate using the DT. They argue that certain contexts are more likely to contain non-referential pronouns and count how often various types of pronouns and other words appear in a small context window. The counts are then provided to a machine learner as features. For our version, we added an anaphoricity feature that fired if a pronoun mention's head was among the top five terms of its own C-expansion. While this helped to raise the F_1 score on the recognition of pleonastic occurrences of the pronoun *it* by 2%, the effect on all types of pronouns was marginal. Furthermore, the number of correctly resolved noun mentions decreased inexplicably, which led to overall worse scores on the coreference metrics.

Deictic mentions: They refer to entities outside of the text but in proximity of the speaker, which was possible as the corpus included transcripts of television newscasts. We fear that they are intrinsically difficult to solve without the accompanying images at hand. After all, the system correctly recognizes those mentions as referring, but it tries to find the referent in the current document.

Definite instances: Some entities can be referred by definite noun phrases if they can be assumed to be recognized by the reader, like e.g. *the FBI* (cf. Jurafsky and Martin 2014, p. 712). The system is unaware of this fact and links them to mentions with semantically similar heads.

Understanding of discourse required (12/7). We put all pairs of coreferent mentions into this category for which it is necessary to consider a larger context than a single sentence. Sometimes this context exceeds even the document itself: One error concerned the mention *our country* in a Chinese news article, which the system incorrectly linked to *Taiwan*. Other errors from this category can be solved by modeling topic shifts or by detecting the boundaries of quoted speech.

Gold error (12/4). In some cases, the system correctly linked two coreferent mentions, yet in the gold document, one of them was either not annotated or its span did not cover the full NP. This should not surprise in light of the size of the corpus and the inter-annotator agreement of 91.8% (Hovy et al. 2006).

Parsing (11/11). Either errornous syntax trees produced by the parser that generated the AUTO documents or the Berkeley system's mention detection algorithm led to an incorrect span selected for a mention that was otherwise resolved correctly.

Syntactic clues (10/3). The only features included by the Berkeley system considering syntax are syntactic uni- and bigrams, however some simple errors can be avoided if one adds features exclusively dealing with pairs of mentions from the same sentence. For example, a mention in subject position cannot be coreferent with a pronoun in object position of the same verb unless it is a reflexive, e.g. the pronoun *them* cannot refer to *the goverment* in the sentence *the government warned them*.

Missing confidence (6/6). The system considered a mention non-anaphoric, even though strong indicators like string matching or definiteness suggest that the system would do so. This also includes our distributional features. For instance, the system did not link the bolded mentions in the sentences[27]

(13) **That period** was both a happy and a fun time. ... Um, of course we say, at **that time**, um, in that class of ours, they weren't all necessarily bad kids.

even though it had positive evidence from the attribute, prior expansion and IS-A features.

Salience (3/4). Likely another side effect of the mention-pair model, the system tends to select other pronouns as antecedents for pronoun mentions, even if they are separated by several sentences, e.g. it links a *she* to a previous *she*, even though a new person was introduced between those two mentions. This ignores the phenomenon of *salience*: in most cases a pronoun refers to a noun mention in one of the previous two sentences (cf. Jurafsky and Martin 2014, p. 713).

Clashing attributes (3/4). Two mentions with identical heads were marked as coreferent even though they had mutually exclusive modifiers, e.g. *the west side* and *the east side*. To address this problem, we modified the instances *from X to Y* and *either X or Y* of the patterns of incompatibility, which are introduced in Section 8.6, to capture adjectives as well. We then added a feature that fired when any pair of attributes from both mentions was reported as incompatible. This feature had no significant effect on the system's metric scores.

Weights (0/4). We found some C-expansion ranks from the interval $[0, 20]$, which we did not discretize, to have unproportionally high ranks assigned to them. The same was true for many lexicalized IS-A features. The reason for this was that the system encountered too few examples of these feature values to learn reasonable weights.

Mention nesting (2/9). A spurious mention contained in another mention, whereby both refer to the same entity. We frequently observed this error for mentions containing an appositive. With its default setting, the Berkeley system predicts both the complete NP and the NP with the appositive removed as mentions that are considered for resolution. For example, it suggests the two mentions denoted by brackets in *[[Kysor], a maker of heavy–duty truck and commercial refrigeration equipment]*. However, only the larger mention is a valid one, since the span should always be expanded to cover the entire NP. While the system learns a feature to avoid linking nested mentions directly, a nesting error occurs if both mentions are resolved to previous mentions from the same entity. Since our attribute features target the appositive, our system linked the smaller mention more frequently. We assume this mechanism to exist because of the high recall approach to mention prediction by Durrett and Klein (2013), as

[27] Document *phoenix_0000* in the development set.

errors in syntax parsing may lead to mentions not being detected. We disabled this behaviour as a test, but noticed a degrade in mention detection and overall performance.

Semantics: recall (8/5) and precision (7/23). This class subsumes all remaining errors that could be related to our semantic features. A recall error occurs if neither the DT, nor the IS-As or the contextual expansion provided the semantic information necessary to resolve the mention. For example, the system was unable to establish that *the pop idol* corefers with *Michael Jackson*, since none of the heads was contained in the prior expansion of the other. We also observed some links like *he – the Hague Tribunal* which could be avoided using animacy data.

A precision error occurs if the system links a mention to the wrong antecedent because one or more semantic features fired. This error type represents no less than 23% of all analyzed errors not made by the baseline. The majority of these errors can be ascribed to the fact that the semantic similarity in a DT's expansion is not restricted to relations compatible with coreference. This results in incorrect bridges like *the man – the woman* or *the road – the sidewalk*.

We found only one error for which the IS-IS-A feature fired: a link between *the country* and *the Chaidamu Basin*. It arose from the fact that one of the sense clusters of *Basin* contains countries like *Mexico* or *Guinea*.

Precision errors that can be attributed to the C-expansion were either coreferent pronouns linked to the selectionally less preferred antecedent, or non-coreferent pronouns being resolved. For an example of the latter, consider the sentence

(14) Thank **you** for your visit and your comment.

The bolded *you* was linked to a previous occurrence of the word *God* by the system since it was ranked first in the pronoun's C-expansion, presumably due to the common phrase *thank God*. The pronoun actually referred to the text's recipient in the gold document, and thus should not have been annotated at all.

In summary, the noise in our features derived from distributional semantics is responsible for 23% of the newly introduced errors, and led to the system's decline in precision. It remains open whether future research is able to produce DTs specialized in the task, or if other distributional approaches like patterns of coreference (Haghighi and Klein 2009) or news story alignment (Recasens et al. 2013) may be a better choice.

8.5 Automatic error classification

A manual analysis of the full error set is impractible. The Berkeley Coreference Analyser (Kummerfeld and Klein 2013)[28] allows to automatically classify different kinds of errors made by a resolution system. It gradually alters, adds or removes entities until the system output equals the gold standard. It defines the following error types, determined by the transformation steps taken to correct the entities:

Span error: A system mention's boundaries were incorrect, either because it did not cover the complete NP or because it included unrelated words.

Conflated entity: Two unrelated entities were considered coreferent by the system.

Divided entity: The system split up a gold entity.

Extra entity: The system produced a spurious entity.

Missing entity: An entire entity is missing from the system's output.

Extra mention: A mention annotated by the system was either a singleton or non-coreferent.

Missing mention: The system deemed a mention not coreferent when in fact it was.

The same system entity may account for multiple errors. For example, an entity which contains both two unrelated key entities and a singleton mention yields one "extra mention" and one "conflated entity" error.

Figure 10 shows the number of errors in each category reported by the tool before and after the addition of our DT-based features. As was expected from better bridging resolution, the number of divided entities decreased.

[28] http://code.google.com/p/berkeley-coreference-analyser/

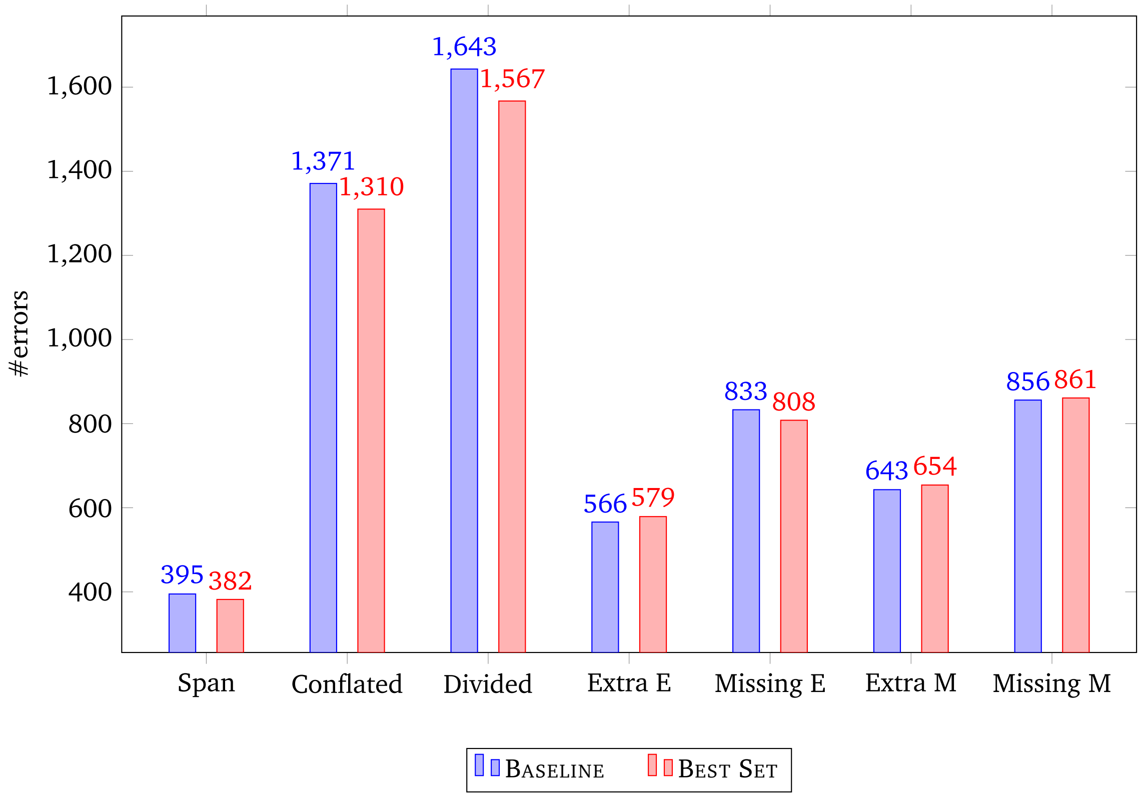


Figure 10: Different types of errors made by the Baseline and Best Set systems on the CoNLL-2011 development set, classified using the Berkeley Coreference Analyser (Kummerfeld and Klein 2013).

Subsequently, the number of conflated entities decreased as well, since both errors frequently occur together (Kummerfeld and Klein 2013). When there is not enough evidence that a mention starts a new entity, the system may falsely link it to the closest predecessor, like in the following case:[29]

(15) Therefore, Hong Kong's Tian Tan Giant Buddha at the most southern tip of China, [...] and Longmen Giant Buddha in the Central Plains, exist as **the representative Buddhas for each of the five directions of China**. As a result, **this place** has become holy land in the hearts of male and female devotees.

By this, *[this place]* is both cut off from its true entity and conjoined with the unrelated *Buddhas*.

Compared to the baseline, Best Set got worse with respect to the number of extra entities, as well as extra and missing mentions. We counted 22 bridging mentions to be responsible for the emergence of extra entities. 8 of them resulted from undesired semantic relations like relational antonymy (*White House – Kremlin*) or metonymy (*MSNBC – the show*) stored in the DT. Regarding the other extra entity errors, we performed a spot-check analysis and noted that the majority of these errors were nested mentions, non-referential pronouns, and mentions with clashing attributes. The system seems also to be biased towards linking mentions with no head match even if none of the semantic features had a positive value.

Regarding extra mentions and missing mentions, Kummerfeld and Klein (2013) argue that they are two sides of the same coin in terms of precision and recall. Systems which decrease the number of the former are prone to increase the number of the latter category, and vice versa. Our system, however, got worse in both categories. In our spot-check analysis, we found a large number of attribute errors being responsible for the extra mentions, which for example resulted in a link between *Hong Kong Wetland Park* and *Wetland Park Exploration Center*. The number of missing mentions increased due to our system having less confidence in linking coreferent mentions with matching heads: compared to the baseline, the pairwise recall in this category dropped from 88% to 86%.

8.6 Trying to overcome the precision barrier: Patterns of incompatibility

In Section 4.1 on Lin's thesaurus we argued that a DT's list of similar words contains semantic relations like antonymy or cohyponymy which are actually negative evidence of coreference. This effect hinders the resolution

[29] Document *cctv_0000* in the development set.

system from having high confidence in top ranks reported by the PRIOR expansion feature, which may be a reason why our system's recall on bridging mentions was still fairly low. Furthermore, the system may erronously reject the hypothesis that a mention is not an antecedent in favor of these "bad" semantic relations. Among the 462 incorrect bridges introduced by the Best Set system when running on the development set, there were 20 instances with heads from semantic relations that are typically not corefent and which could be found in each other's prior expansion (Table 8a). Since semantic precision errors are the major problem of our system, the system could profit from better expansion quality, especially with regard to proper nouns, whose similar terms are almost entirely made up of cohyponyms.

		Count
Rita	Katrina	**11**
senate	chamber	0
boy	granddaughter	0
navy	airforce	**2**
brother	wife	**4**
Dan	Linda	0
MSNBC	show	0
Shanghai	Beijing	**219**
ship	navy	0
Serbia	city	0
pilot	attendant	**2**
country	world	**2**
season	year	**3**
day	decade	**1**
government	party	**39**
today	August	**1**
Senate	committee	0
country	Europe	**11**
Kremlin	(White) House	0
yesterday	night	0

(a)

		Count
man	woman	4240
India	Pakistan	3724
Israelis	Palestinians	2359
Palestinians	Israel	2331
Republicans	Democrats	1598
China	India	1443
China	Japan	1443
House	Senate	1281
west	east	1255
China	Taiwan	1128
head	toe	1069
south	north	1066
Muslims	christian	1031
government	rebel	1019
death	life	930
Germany	France	911
Israel	Hezbollah	884
Israel	Syria	867
church	state	787
blacks	whites	752

(b)

Table 8: Lists of pairs of incompatible words and their frequency in the list automatically acquired from a 120M sentences corpus. Left: Head words of spurious bridges introduced by the Best Set system. Right: 20 most frequent pairs in the antonym data.

To automatically acquire a list of antonymous words, we make use of *patterns of incompatibility* as introduced by Lin et al. (2003), who use them to identify synonyms among the expansion of a DT. Those Hearst-style patterns mostly match pairs of words with opposing meanings. To test whether two words are truly synonymous, they are inserted into each pattern, which is then submitted to a web search engine. The words are considered compatible if the search shows few results compared to the frequency they appear in close proximity in general.

Asking the web about every term pair of each mention's expansion is impracticable. Instead, we used the set of patterns enumerated in Figure 11 to extract incompatible words directly from the same 120M sentences corpus the DT was computed on. Inevitably, this naïve approach suffers from sparsity problems.

As we wanted to rule out cohyponyms as well, we added two patterns derived from IS-A counterparts: *such as* and *including*. They match a varying number of words (not only two), so each possible pair of non-identical words from this set was considered incompatible. To match the DT's term representation, all words were lemmatized and annotated with POS tags using the Stanford CoreNLP pipeline (Manning et al. 2014). Only single nouns, optionally preceded by determiners, were regarded by us to be admissable pattern fillers. Without this restriction, valid synonyms like for example found in *from romantic flicks to horror movies* would have been matched as well. Unfortunately, this measure greatly reduced the already small number of obtained words, leaving us with only 350K distinct pairs. It also prevents us from finding multiword expressions like *White House*. Table 8b lists the 20 pairs observed most frequently.

1. from X to Y
2. either X or Y
3. such as $(X_i,)^*$ (and|or (X_i))
4. including $(X_i,)^*$ (and|or (X_i))
5. X, in contrast to Y
6. X vs. Y
7. X, Y and (other|similar)
8. between X and Y
9. battle of X and Y
10. X *<be>* *<comparative>* than Y
11. X, unlike Y

Figure 11: The set of patterns used to acquire incompatible words. The first two are from Lin et al. (2003).

As mentioned earlier, syntagmatic approaches like pattern matching suffer from low recall. The obtained set of incompatible words is far to sparse to completely clear a prior expansion of unwanted terms: only 11 of the 20 pairs of wrongly bridging mention heads listed in Table 8 had a non-zero count. Unfortunately, we cannot apply the clustering technique that was used in the JoBimText DT to counter IS-A sparsity Gliozzo et al. (2013), since we want to separate those terms instead of merging them.

The antonym data was used to augment the coreference resolution system in two ways: We modifed a term's prior expansion to exclude all terms it is incompatible with, and we introduced a feature INCOMP whose value is true if the heads of the current and preceding mention are incompatible. Two terms are considered incompatible if the number of times they occurred together in patterns of incompatibility during the data acquisition phase exceeds a threshold k. The parameter is necessary since the patterns return some false positive pairs like <*president, man*> in rare cases. We set k to 4 for our database. The value was determined by incrementally increasing it until the amount of coreferent mentions with noun heads incorrectly identified as incompatible in the training set was below 1%. Table 9 shows how the incompatibility filter and feature affects the metric scores as well as precision and recall of bridging mentions in comparison to the Best Set system.

Metric scores				Bridging mentions		
MUC-F_1	B^3-F_1	CEAF$_e$-F_1	Avg	P	R	F_1
-0.14	-0.14	-0.07	-0.11	-0.56	± 0	-0.19

Table 9: Change in metric scores and precision/recall on bridging mentions when adding automatically acquired antonym data to the Best Set system, evaluated on the CoNLL-2011 development set.

Not only did all metric scores decrease, but there is also a considerable decline in precision on bridging mentions, while their recall remains unchanged. It can be assumed from this that the system assigned larger weights to high ranks in the purged prior expansion, as was intended, but the actual quality of those top terms did not change for the better. This is because by removing terms from the DT's prior expansion, we disrupt the symmetry originally imposed by the similarity measure. For every "bad" term removed, others not recalled by our patterns rise in rank and will therefore be considered semantically more similar. Table 10 illustrates this problem: While we were able to remove incompatible terms like *woman* and *girl*, we now see *grandmother* and *housewife* being ranked higher up. Also note that *gunman* was incorrectly removed as well, indicating that our patterns are not well selected in terms of precision.

woman	person
teenager	male
person	businessman
male	**grandmother**
businessman	**hosewife**
girl	suspect
boy	robber
gunman	frenchman
grandmother	guy
housewife	drifter
...	...

Table 10: Prior expansion of common noun *man* before (left) and after (right) applying the filter. Incompatible terms are marked in bold.

8.7 Comparison to a context window holing operation

Up to this point, we trained and evaluated the coreference resolution system using always the same DT based on dependency relations. This section investigates how much the choice of holing system influences the system's performance. We tried out an additional DT provided by the JoBimText Project which uses a context window of size 3 with the hole resp. mention head being in its center, e.g. an example term-context pair is *(Obama, President @ said)*. The terms are the unaltered surface tokens. The DT was created from the same 120M sentences data basis as the dependency DT. The contexts were taken only from the same sentence as the term, so any out-of-sentence token was replaced by a placeholder.

The dependency and context window holing operations are compared with the features PRIOR, SHARED_PRIOR and ATTR_PRIOR enabled. Apart from those, no additional feature based on distributional semantics has been added to the Berkeley Final feature set. We did not include the other DT-centric feature, context-based expansion, for a number of reasons. First of all, since the context features include a term's preceding token, common nouns tend to not show up in the expansion if the determiner is missing from the context. To elaborate this further, consider the term-context pair *(she, when @ barked)*. Because of the verb *bark*, we expect encountering *dog* in the context's C-expansion, though the pair *(dog, when @ barked)* is not registered in the DT since the determiner is missing. To account for this, we would have to mix in three copies of the term's context feature for which the first element had been replaced with one of the determiners *a*, *an* and *the*. However, with 3 out of 4 context features starting with determiners, the expansion would now be biased towards common nouns. Our second caveat against using the C-expansion is that most of the time, the verb, being the most informative part of the sentence, is not observed in direct neighborhood of a mention's head.[30] The expansion of a small context window of size 3 is therefore inadequate to model selectional preference. Last but not least, even if these problems are neglected, the computation time increased vastly because of many common but uninformative context features like *the @ which*, yielding long lists of terms to weight and sort.

Like in the preceding sections, the systems are being judged by consulting the coreference metric scores as well as precision, recall and F_1 on pairwise links and bridging mentions. The results on the development set are shown in Table 11 and compared to a system with a dummy thesaurus.

	Metrics				Pairwise linkage			Bridging mentions		
	MUC-F_1	B^3-F_1	CEAF$_e$-F_1	Avg	P	R	F_1	P	R	F_1
Dummy-DT	66.18	57.21	56.67	60.02	60.85	60.01	60.42	35.59	20.87	26.31
Dep	66.28	57.26	56.83	60.12	**70.96**	**69.40**	**70.17**	**37.87**	**24.93**	**30.01**
CW	**66.50***	**57.55***	**56.87**	**60.31***	70.90	69.21	70.05	37.30	23.18	28.59

Table 11: Comparison of a depencency parse holing system (Dep) to one using a context window (CW). Only prior expansion and attribute features were enabled. Starred coreference metric scores are significant improvements over the system running on a dummy thesaurus.

The dummy setting produced almost identical scores for the two holing systems, which is why we reported only those obtained with the dependency-based operation in Table 11.[31] While the context window operation seems to perform better than the one using dependency relations, this might be due to chance. We could only establish a statistically significant improvement of MUC and B^3 recall between the two systems. When examining pairwise linkage counts, we found that the syntactic holing system performs slightly worse on anaphoric pronouns and non-bridging nominals. Since both types make up about 64% of coreferent mentions, missing one may lead to divided entities. We assume the discrimination between common and proper noun terms in the DT is responsible for the decreased number of resolved coreferent mentions with matching heads. For example, *service#NN*, the common noun term, cannot be found among the expansion of *Service#NP*, the proper noun term. As a consequence, POS tagging errors caused the syntactic DT to have 4.7 times as many out-of-vocabulary errors (i.e. terms without prior expansions) as the context window DT.

The results on bridging mentions support the intuition that dependency relations as context features lead to a better thesaurus quality than a small context window, which is also backed by experiments by Biemann and

[30] In the CoNLL-2011 training set, only 28.41% of the heads of predicted mentions were preceded or followed by a verb.

[31] Differences between the score here and the one in Section 8.2 above are due to the reduced feature set. Albeit the rest of the features have the same value for all pairs of mentions in a dummy setting, the machine learner will assign non-zero weights to their feature conjunctions, as they convey valuable information of coreference.

Riedl (2013b). However, Agirre et al. (2009) showed that a broader window of 9 tokens produces better semantic simlarity scores than a model with syntactic context features, at least when trained on a huge (Teraword-sized) corpus.

9 Conclusion and future work

We have shown that features derived from distributional semantics are able to improve a state-of-the-art coreference resolution system. We resolved an increased number of bridging mentions compared to the baseline, but because of the system's lack of confidence in our features, their recall is still low. However, a loss in overall resolution precision remains the biggest challenge, and countermeasures against it should be the main focus of future research in the topic. A distributional thesaurus contains many similar words with unwanted semantic relations, which we were unable to remove using automatically mined pairs of incompatible terms. Ideally, the similar words in the thesaurus expansion are separated by their semantic relation to the target term. A possible solution are directional similarity measures which allow to bring words of the desired relation, e.g. hypernyms or antonyms, to the expansion's top (cf. Lenci 2014).

The Berkeley system uses a mention-pair model that weights each pair of mentions independently. Not only did this give rise to nested mentions being marked as coreferent, but it also prevented us from directly using the semantic similarity score between two mentions. Ranking models, on the other hand, compare all possible antecedents and are therefore ideal candidates for our rank-based features. Context-based expansion in particular could profit from a direct comparison of ambiguous antecedents.

There is much room for experiments regarding the holing operation. For example, like some of the context features used to create a VSM in Lenci (2011), a path of size 2 in the dependency parse could better model selectional restrictions. Since identity coreference deals with noun phrases only, the DT's terms can be restricted to that POS, which not only speeds up the computation but also eliminates the need for POS-tagged terms to distinguish noun/verb homonyms like *fall*. Another possibility is using multiple words from a NP as a term, which should especially be useful to distinguish proper names.

The terms stored in the JoBimText DTs are clustered into senses, yet we did not perform word sense disambiguation. This leads to errors made by our system like in the following toy example:

(16) [Denzel Washington] won **[an Oscar]**. **[The actor]** thanked [his colleagues].

The systems comes to the false conclusion that *[the actor]* refers to *[an Oscar]* because the IS-IS-A feature fired for both predecessing mentions and *[an Oscar]* was closer to the anaphor. By itself, *Oscar* is an entirely legitimate name for an actor, yet in context it becomes clear that the mention references the award of the same name. Gliozzo et al. (2013) suggested that the term's sense in context can be determined using context-sensitive re-ranking. This way, we are able to consider only the cluster which contains award-related IS-As.

We believe that in the long term, knowledge which was obtained automatically from unannotated corpora will be superior to any hand-written thesaurus in the task of bridging mentions resolution. While it is already difficult to keep track of the natural changes of language, capturing all the new names of things, persons and organizations that come into the limelight on a daily basis requires a considerable amount of effort, probably even more than humanly possible. Moreover, if a language has only a small number of speakers, there may exist no electronical lexical resources for it at all. It is already expensive to create a high-quality corpus with coreference annotations; one should not have to write a thesaurus as well.

References

Agirre, Eneko et al. (2009). “A Study on Similarity and Relatedness using Distributional and WordNet-based Approaches”. In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL*. Boulder, Colorado, pp. 19–27.

Appelt, Douglas E. (1999). “Introduction to information extraction”. In: *Ai Communications* 12.3, pp. 161–172.

Bagga, Amit 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 (LREC ’98)*. Vol. 1, pp. 563–566.

Baker, Collin F., Charles J. Fillmore, and John B. Lowe (1998). “The Berkeley FrameNet Project”. In: *Proceedings of the COLING-ACL*. Montreal, Canada, pp. 86–90.

Bean, David and Ellen Riloff (2004). “Unsupervised Learning of Contextual Role Knowledge for Coreference Resolution”. In: *Proceedings of the Human Language Technology Conference / North American Chapter of the ACL Annual Meeting 04*, pp. 297–304.

Bergsma, Shane and Dekang Lin (2006). “Bootstrapping Path-Based Pronoun Resolution”. In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*. Sydney, Australia: ACL, pp. 33–40.

Bergsma, Shane, Dekang Lin, and Randy Goebel (2008a). “Discriminative learning of selectional preference from unlabeled text”. In: *Proceedings of EMNLP 08*. Honolulu, Hawaii: ACL, pp. 59–68.

– (2008b). “Distributional Identification of Non-Referential Pronouns”. In: *ACL-HLT 2008*. Columbus, Ohio: ACL, pp. 10–18.

Biemann, Chris (2006). “Chinese Whispers: an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems”. In: *Proceedings of the HLT-NAACL-06 Workshop on Textgraphs-06*. ACL. New York, USA, pp. 73–80.

Biemann, Chris and Martin Riedl (2013a). “From Global to Local Similarities: A Graph-Based Contextualization Method using Distributional Thesauri”. In: *Proceedings of the 8th Workshop on TextGraphs in conjunction with EMNLP 2013*. Seattle, WA, USA.

– (2013b). “Text: Now in 2D! A Framework for Lexical Expansion with Contextual Similarity”. In: *Journal of Language Modelling* 1.1, pp. 55–95.

Biemann, Chris et al. (2008). “ASV Toolbox: a Modular Collection of Language Exploration Tools”. In: *Proceedings of LREC-08*. Marrakesh, Marocco.

Calvo, Hiram, Kentaro Inui, and Yuji Matsumoto (2009). “Learning Co-relations of Plausible Verb Arguments with a WSM and a Distributional Thesaurus”. In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Berlin and Heidelberg: Springer, pp. 363–370.

Caraballo, Sharon A. and Eugene Charniak (1999). “Determining the specificity of nouns from text”. In: *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. College Park, MD, USA, pp. 63–70.

Clark, Herbert H. (1975). “Bridging”. In: *Proceedings of the 1975 workshop on Theoretical issues in NLP*. ACL. New Haven, CT, USA, pp. 169–174.

Curran, James R. and Marc Moens (2002). “Improvements in Automatic Thesaurus Extraction”. In: *Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon*. ACL. Philadelphia, USA, pp. 59–66.

Dagan, Ido and Alon Itai (1990). “Automatic Processing of Large Corpora for the Resolution of Anaphora References”. In: *Proceedings of the 13th conference on Computational linguistics*. Vol. 3. ACL. Helsinki, Finland, pp. 330–332.

van Deemter, Kees and Rodger Kibble (2000). “On Coreferring: Coreference in MUC and Related Annotation Schemes”. In: *Computational Linguistics* 26.4, pp. 629–637.

Dice, Lee R. (1945). “Measures of the Amount of Ecologic Association Between Species”. In: *Ecology* 26.3, pp. 297–302.

Durrett, Greg, David Hall, and Dan Klein (2013). "Decentralized Entity-Level Modeling for Coreference Resolution". In: *Proceedings of the 51st Annual Meeting of the ACL (Volume 1: Long Papers)*. Sofia, Bulgaria: ACL, pp. 114–124.

Durrett, Greg and Dan Klein (2013). "Easy Victories and Uphill Battles in Coreference Resolution". In: *Proceedings of the Conference on EMNLP*. Seattle, Washington: ACL.

Erk, Katrin (2007). "A Simple, Similarity-based Model for Selectional Preferences". In: *Proceedings of ACL 2007*. Vol. 45. 1. Prague, Czech Republic, pp. 216–223.

Erk, Katrin and Sebastian Padó (2008). "A Structured Vector Space Model for Word Meaning in Context". In: *Proceedings of EMNLP 2008*. ACL. Honolulu, Hawaii, pp. 897–906.

Erk, Katrin, Sebastian Padó, and Ulrike Padó (2010). "A Flexible, Corpus-Driven Model of Regular and Inverse Selectional Preferences". In: *Computational Linguistics* 36.4, pp. 723–763.

Evert, Stefan (2005). "The Statistics of Word Cooccurrences. Word Pairs and Collocations". PhD thesis. Stuttgart University.

Gasperin, Caroline, Susanne Salmon-Alt, and Renata Vieira (2004). "How useful are similarity word lists for indirect anaphora resolution?" In: *Proceedings of DAARC 2004*. Sao Miguel, Azores.

Gliozzo, Alfio et al. (2013). "JoBimText Visualizer: A Graph-based Approach to Contextualizing Distributional Similarity". In: *Proceedings of the 8th Workshop on TextGraphs in conjunction with EMNLP 2013*. Seattle, WA, USA.

Gubbins, Joseph and Andreas Vlachos (2013). "Dependency Language Models for Sentence Completion". In: *Proceedings of EMNLP '13*. Seattle, USA, pp. 1405–1410.

Haghighi, Aria and Dan Klein (2009). "Simple Coreference Resolution with Rich Syntactic and Semantic Features". In: *Proceedings of EMNLP '09*. EMNLP '09. Singapore: ACL, pp. 1152–1161.

Harris, Zellig S. (1954). "Distributional Structure". In: *Word* 10.2-3, pp. 146–162. Reprint in Zellig, S. Harris (1970): *Papers in Structural and Transformational Linguistics*, Reidel: Dordrecht, pp. 775-794.

Hearst, Marti A. (1992). "Automatic Acquisition of Hyponyms from Large Text Corpora". In: *Proceedings of the 14th conference on Computational linguistics-Volume 2*. ACL. Nantes, France, pp. 539–545.

– (1998). "Automated Discovery of WordNet Relations". In: *WordNet: An Electronic Lexical Database*. Ed. by Christiane Fellbaum. MIT Press, pp. 131–151.

Hindle, Donald (1990). "Noun Classification from Predicate-Argument Structures". In: *Proceedings of the 28th meeting of the ACL*. ACL, pp. 268–275.

Hirschman, Lynette and Nancy Chinchor (1997). "MUC-7 Coreference Task Definition". In: *Proceedings of the 7th Message Understanding Conference (MUC-7)*. URL: `http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/co_task.html`.

Hovy, Eduard et al. (2006). "OntoNotes: The 90% Solution". In: *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*. ACL, pp. 57–60.

Jurafsky, Daniel and James H. Martin (2014). *Speech and Language Processing*. 2nd ed. Essex: Pearson.

Kerber, Randy (1992). "Chimerge: Discretization of Numeric Attributes". In: *Proceedings of the 10th national conference on Artificial Intelligence*. AAI Press. San Jose, California, pp. 123–128.

Kobdani, Hamidreza et al. (2011). "Bootstrapping Coreference Resolution Using Word Associations". In: *Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies - Volume 1*. Portland, Oregon: ACL, pp. 783–792.

Koehn, Philipp (2004). "Statistical Significance Tests for Machine Translation Evaluation". In: *Proceedings of EMNLP 2004*, pp. 388–395.

Kotlerman, Lili et al. (2010). "Directional distributional similarity for lexical inference". In: *Natural Language Engineering* 16.4, pp. 359–389.

Kummerfeld, Jonathan K. and Dan Klein (2013). "Error-Driven Analysis of Challenges in Coreference Resolution". In: *Proceedings of EMNLP*. Seattle, WA, USA.

Kunz, Kerstin Anna (2010). *Variation in English and German Nominal Coreference. A Study of Political Essays*. Saarbrücker Beiträge zur Sprach- und Translationswissenschaft 21. Frankfurt am Main et al.: Peter Lang.

Lee, Heeyoung et al. (2011). “Stanford’s Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task”. In: *Proceedings of the Fifteenth Conference on CoNLL: Shared Task*. ACL, pp. 28–34.

Lee, Heeyoung et al. (2012). “Joint Entity and Event Coreference Resolution across Documents”. In: *Proceedings of the 2012 Joint Conference on EMNLP and CoNLL*. ACL, pp. 489–500.

Lee, Heeyoung et al. (2013). “Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules”. In: *Computational Linguistics* 39.4, pp. 885–916.

Lee, Lillian (1999). “Measures of Distributional Similarity”. In: *Proceedings of the 37th meeting of the ACL*, pp. 25–32.

Lenci, Alessandro (2011). “Composing and Updating Verb Argument Expectations: A Distributional Semantic Model”. In: *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*. ACL. Portland, pp. 58–66.

– (2014). “Will Distributional Semantics Ever Become Semantic?” Script of a talk at the 7th International Global WordNet Conference. URL: `gwc2014.ut.ee/lenci_distributional_semantics_gwc2014.pdf`.

Lin, Dekang (1998). “Automatic Retrieval and Clustering of Similar Words”. In: *Proceedings of the 17th International Conference on Computational Linguistics*. Vol. 2. Montreal, Quebec, Canada: ACL, pp. 768–774.

Lin, Dekang et al. (2003). “Identifying Synonyms among Distributionally Similar Words”. In: *Proceedings of IJCAI-03*. Vol. 3, pp. 1492–1493.

Luo, Xiaoqiang (2005). “On Coreference Resolution Performance Metrics”. In: *Proceedings of the conference on Human Language Technology and EMNLP ’05*. ACL. Vancouver, Canada, pp. 25–32.

Manning, Christopher D. and Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge (Massachusetts): MIT press.

Manning, Christopher D. et al. (2014). “The Stanford CoreNLP Natural Language Processing Toolkit”. In: *Proceedings of 52nd Annual Meeting of the ACL: System Demonstrations*, pp. 55–60.

Marneffe, Marie-Catherine de, Bill MacCartney, and Christopher D. Manning (2006). “Generating Typed Dependency Parses from Phrase Structure Parses”. In: *Proceedings of LREC*. Vol. 6, pp. 449–454.

McCarthy, Diana and Roberto Navigli (2009). “The English lexical substitution task”. In: *Language Resources & Evaluation* 43.2, pp. 139–159.

Miller, George A. (1995). “WordNet: a Lexical Database for English”. In: *Communications of the ACM* 38.11, pp. 39–41.

Mitkov, Ruslan (2002). *Anaphora Resolution*. London et al.: Pearson.

Ng, Vincent (2007). “Shallow Semantics for Coreference Resolution”. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 1689–1694.

– (2010). “Supervised Noun Phrase Coreference Research: The First Fifteen Years.” In: *Main Proceedings of the 48th annual ,eeting of the ACL*, pp. 1396–1411.

Padó, Sebastian and Mirella Lapata (2007). “Dependency-Based Construction of Semantic Space Models”. In: *Computational Linguistics* 33.2, pp. 161–199.

Parker, Robert et al. (2011). *English Gigaword Fifth Edition LDC2011T07*. Web Download. Philadelphia: Linguistic Data Consortium. URL: `http://catalog.ldc.upenn.edu/LDC2011T07`.

Picard, Justin (1999). “Finding content-bearing terms using term similarities”. In: *Proceedings of the 9th conference on European chapter of the ACL*. ACL, pp. 241–244.

Poesio, Massimo, Sabine Schulte im Walde, and Chris Brew (1998). “Lexical Clustering and Definite Description Interpretation”. In: *Proceedings of the AAAI Spring Symposium on Learning for Discourse*, pp. 82–89.

Poesio, Massimo et al. (2004). “Learning to Resolve Bridging References”. In: *Proceedings of the 42nd Annual Meeting of the ACL*. ACL.

Ponzetto, Simone Paolo and Michael Strube (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 ACL*. ACL. New York, New York, USA, pp. 192–199.

Pradhan, Sameer et al. (2011). “CoNLL-2011 Shared Task: Modeling Unrestricted Coreference in OntoNotes”. In: *Proceedings of the Fifteenth Conference on Computational Natural Language Learning (CoNLL 2011)*. Portland, Oregon, pp. 1–27.

Pradhan, Sameer et al. (2014). “Scoring Coreference Partitions of Predicted Mentions: A Reference Implementation”. In: *Proceedings of the 52nd Annual Meeting of the ACL*. Baltimore, MD, USA, pp. 22–27.

Raghunathan, Karthik (2010). *Simple Coreference Resolution with Rich Syntactic and Semantic Features: Is it enough?* Research report. Stanford, California: Stanford University.

Rahman, Altaf and Vincent Ng (2009). “Supervised Models for Coreference Resolution”. In: *Proceedings of the 2009 Conference on EMNLP*. Vol. 2. ACL, pp. 968–977.

– (2011). “Coreference resolution with world knowledge”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, pp. 814–824.

Recasens, Marta, Matthew Can, and Daniel Jurafsky (2013). “Same Referent, Different Words: Unsupervised Mining of Opaque Coreferent Mentions”. In: *Proceedings of HLT-NAACL*, pp. 897–906.

Resnik, Philip (1996). “Selectional constraints: An information-theoretic model and its computational realization”. In: *Cognition* 61.1, pp. 127–159.

Richter, Matthias et al. (2006). “Exploiting the Leipzig Corpora Collection”. In: *Proceesings of the IS-LTC 2006*.

Riedl, Martin and Chris Biemann (2013). “Scaling to Large^3 Data: An Efficient and Effective Method to Compute Distributional Thesauri”. In: *Proceedings of EMNLP 2013*, pp. 884–890.

Rubenstein, Herbert and John B. Goodenough (1965). “Contextual correlates of synonymy”. In: *Communications of the ACM* 8.10, pp. 627–633.

Sahlgren, Magnus (2006). “The Word-Space Model. Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces”. PhD thesis. Stockholm University.

de Saussure, Ferdinand (1931/2001). *Grundfragen der allgemeinen Sprachwissenschaft*. German. Ed. by Charles Bally and Albert Sechehaye. Trans. from the French by Herman Lommel. 3rd ed. Berlin and New York: de Gruyter.

Shaw Jr, W. M., Robert Burgin, and Patrick Howell (1997). “Performance Standards and Evaluations in IR Test Collections: Cluster-based Retrieval Models”. In: *Information Processing & Management* 33.1, pp. 1–14.

Shi, Shuming et al. (2010). “Corpus-based Semantic Class Mining: Distributional vs. Pattern-Based Approaches”. In: *Proceedings of the 23rd International Conference on Computational Linguistics*. ACL. Beijing, China, pp. 993–1001.

Soon, Wee Meng, Hwee Tou Ng, and Daniel Chung Yong Lim (2001). “A Machine Learning Approach to Coreference Resolution of Moun Phrases”. In: *Computational Linguistics* 27.4, pp. 521–544.

Versley, Yannick (2007). “Antecedent Selection Techniques for High-Recall Coreference Resolution”. In: *Proceedings of the 2007 Joint Conference on EMNLP and CoNLL*. ACL. Prague, pp. 496–505.

Vieira, Renata and Massimo Poesio (2000). “An Empirically Based System for Processing Definite Descriptions”. In: *Computational Linguistics* 26.4, pp. 539–593.

Vieira, Renata and Simone Teufel (1997). “Towards resolution of bridging descriptions”. In: *Proceedings of the 35th Annual Meeting of the ACL and Eighth Conference of the European Chapter of the ACL*. ACL, pp. 522–524.

Vilain, Marc et al. (1995). “A model-Theoretic Coreference Scoring Scheme”. In: *Proceedings of the 6th conference on Message Understanding*. ACL, pp. 45–52.

Yang, Xiaofeng, Jian Su, and Chew Lim Tan (2005). “Improving Pronoun Resolution Using Statistics-Based Semantic Compatibility Information”. In: *Proceedings of the 43rd Annual Meeting of the ACL*. ACL, pp. 165–172.

Zhou, Huiwei et al. (2011). “Combining Syntactic and Semantic Features by SVM for Unrestricted Coreference Resolution”. In: *Proceedings of the Fifteenth Conference on CoNLL: Shared Task*. ACL, pp. 66–70.