Natural Language Engineering **1** (1): 1–29. Printed in the United Kingdom

Exploiting the Wikipedia Structure in Local and Global Classification of Taxonomic Relations†

QUANG XUAN DO and DAN ROTH

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
emails: quangdo2@illinois.edu, danr@illinois.edu

(*Received 30 April 2011; revised 28 September 2011; accepted 7 December 2011*)

Abstract

Determining whether two terms have an ancestor relation (e.g. *Toyota Camry* and *car*) or a sibling relation (e.g. *Toyota* and *Honda*) is an essential component of textual inference in Natural Language Processing applications such as Question Answering, Summarization, and Textual Entailment. Significant work has been done on developing knowledge sources that could support these tasks, but these resources usually suffer from low coverage, noise, and are inflexible when dealing with ambiguous and general terms, that may not appear in any stationary resource, making their use as general purpose background knowledge resources difficult. In this paper, rather than building a hierarchical structure of concepts and relations, we describe an algorithmic approach that, given two terms, determines the taxonomic relation between them using a machine learning-based approach that makes use of existing resources. Moreover, we develop a global constraint-based inference process that leverages an existing knowledge base to enforce relational constraints among terms and thus improves the classifier predictions. Our experimental evaluation shows that our approach significantly outperforms other systems built upon existing well-known knowledge sources.

1 Introduction

Fundamental taxonomic relations such as *ancestor-descendant* (e.g. *actor* and *Mel Gibson*) and *siblings* (e.g. *Mel Gibson* and *Tom Cruise*) have been shown to hold important roles in many computational linguistics tasks, such as document clustering (Hotho *et al.*, 2003), navigating text databases (Chakrabarti *et al.*, 1997),

† This research is supported by the Army Research Laboratory (ARL) under agreement W911NF-09-2-0053 and the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181. The first author also thanks the Vietnam Education Foundation (VEF) for its sponsorship. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the VEF, ARL, DARPA, AFRL, or the US government.

Question Answering (QA) (Saxena *et al.*, 2007) and Summarization (Vikas *et al.*, 2008). Recently, it has been shown that recognition of taxonomic relations between terms is essential to support textual inference tasks such as Textual Entailment (TE) (Dagan *et al.*, 2006). For example, it may be important to know that a *blue Toyota Prius* is neither a *white Toyota Prius* nor a *blue Toyota Camry*, and that all are *compact cars*. Work in TE has argued quite convincingly (MacCartney and Manning, 2008) that many such textual inferences are largely compositional and depend on the ability to recognize fundamental taxonomic relations, such as the ancestor or sibling relations, between terms. Furthermore, several TE studies (Abad *et al.*, 2010; Sammons *et al.*, 2010) suggest isolating TE phenomena, including recognizing taxonomic relations, and studying them separately. They also discuss characteristics of several phenomena (e.g. contradiction) from a perspective similar to ours, but do not provide a solution.

Motivated by the needs of natural language processing tasks, and the compositionality argument alluded to above, this paper addresses the problem of classifying fundamental taxonomic relations between terms: given two well-segmented terms, the system predicts the taxonomic relation between them – *ancestor-descendant*, *siblings* or *no relation*. In this work, the context where the terms come from is not given. We leave the idea of leveraging the context of the input terms and how to use the taxonomic relations in applications to a future extension of this work.

An input term could be any well-segmented span of words that refers to a concept. Moreover, input terms may include common nouns or proper nouns from open or closed concept classes. Some examples of input terms include *mountain*, *George W. Bush*, *battle of Normandy*, *table*, *US Today*, *NATO*, and *chemical elements*. In this paper, we use *term* and *concept* interchangeably, even though *concept* is usually used to refer to nodes in hierarchical resources. For taxonomic relations, we consider that two terms hold an *ancestor-descendant* relation if one term is subsumed by the other w.r.t. a taxonomic structure, whereas two terms are *siblings* if they share a common subsumer.

An ancestor-descendant relation and its directionality can help us infer that a text snippet mentioning a descendant term (e.g. *cannabis*) entails a hypothesis mentioning an ancestor term (e.g. *drugs*) in a similar way as in the following example, taken from a TE challenge data set.

Text: Nigeria's NDLEA has seized 80 metric tons of *cannabis* in one of its largest ever hauls, officials say.

Hypothesis: Nigeria seizes 80 tons of *drugs.*

Similarly, it is important to know of a sibling relation to infer that a statement about *Taiwan* (without additional information) is not likely to entail a hypothesis about *Japan* since they are different countries, as in the following example:

Text: A strong earthquake struck off the southern tip of *Taiwan* at 12:26 UTC, triggering a warning from Japan's Meteorological Agency that a 3.3 foot tsunami could be heading towards Basco, in the Philippines.

Hypothesis: An earthquake struck *Japan.*

Naturally, these taxonomic relations can be read off from manually generated resources such as Wordnet that explicitly represent these relations. However, it is clear that these resources have limited coverage. For example, Wordnet 3.0 (Fellbaum, 1998) consists of only around 118,000 nominal concepts, which is obviously much smaller than the number of concepts in English. In addition, very few entities and multiword concepts are covered in WordNet.

There has also been work on extending the manually built resources using automatic acquisition methods resulting in structured knowledge bases such as the Extended WordNet (Snow *et al.*, 2006) and the YAGO ontology (Suchanek *et al.*, 2007). These knowledge sources only partially alleviate the coverage problem, and could be potentially impaired by noise introduced when they were compiled.

One of the well-known approaches to building offline resources is using relational patterns (e.g. *X such as Y, Z*) to extract related terms from text (Hearst, 1992; Snow *et al.*, 2006). Unfortunately, this approach is usually brittle. Infrequent terms are less likely to be covered, and may not be effectively extracted since they do not usually appear in close proximity with other terms (e.g. Israeli tennis player *Dudi Sela* and Swiss tennis champion *Roger Federrer* rarely appear together in news text). On the other hand, knowledge sources derived by using bootstrapping algorithms and distributional semantic models (Pantel and Pennacchiotti, 2006; Kozareva *et al.*, 2008; Baroni and Lenci, 2010) typically suffer from a trade-off between precision and recall, resulting either in a relatively accurate resource with low coverage or a noisy resource with broader coverage.

Another limitation of structured resources, as we observe, is their inflexibility in dealing with terms which cannot be exactly mapped to existing concepts in the resources. This problem usually occurs when a resource actually contains a concept corresponding to an input term, but the concept and the term are written with different surface strings. For example, one may not be able to map the input term *Chelsea* to concept *Chelsea, London* (an area of West London) in the Extended WordNet using an exact string-matching operation because their surface strings are not the same. Even worse, if the Extended Wordnet also maintains the concept *Chelsea F.C.* (an English football club based in West London) in addition to *Chelsea, London*, then there is no clear mechanism to map the input term *Chelsea* to the concept *Chelsea, London* or *Chelsea F.C.*[1]

In this paper, we present a novel approach to identifying the taxonomic relation between two input terms by exploiting the rich structure and information of Wikipedia,[2] a free and collaboratively updated encyclopedia of concepts. It is important to emphasize that our work focuses on directly classifying relations that hold between input terms rather than building a resource of relational information among concepts. In this respect, we are distinct from Open Information Extraction (Banko *et al.*, 2007), on-demand Information Extraction (Sekine, 2006), and other

[1] One may write a better coreference/ambiguity resolver to deal with ambiguous terms. However, it is not feasible when the context of the input terms is not given as in this work.

[2] http://wikipedia.org/

efforts to recognize facts in a given corpus (Davidov and Rappoport, 2008; Paşca and Van Durme, 2008), which capitalize on local co-occurrence of terms to generate databases of open-ended facts. Our work is also different from the supervised relation extraction effort (Roth and Yih, 2004) that requires full text or sentences, where the two terms appear, to infer their relation.

In our work, we use Wikipedia as a background knowledge source. This resource has been shown to be very useful and powerful for many tasks in knowledge extraction (Suchanek *et al.*, 2007; Ponzetto and Strube, 2007), information retrieval (Milne and Witten, 2008; Mihalcea and Csomai, 2007; Ratinov *et al.*, 2011), and computing semantic relatedness (Gabrilovich and Markovitch, 2007). One of the most important advantages of Wikipedia is that it allows volunteers to contribute their knowledge collaboratively. Wikipedia, therefore, keeps growing over time with millions of relations and concepts, including common nouns and proper nouns (e.g. *chicken*, *blue*, *Everest*, *US Today*) and open and closed concept classes (e.g. *country*, *foods*, *chemical elements*). Specifically, Wikipedia was chosen for our work for the following reasons:

- Wikipedia consists of millions of pages providing rich information about concepts. The pages in Wikipedia are well organized in an informative structure. This allows us to easily leverage the information in Wikipedia to support classification decisions.
- The information in Wikipedia is collaboratively generated, modified and updated. Volunteers around the world contribute to Wikipedia everyday, guaranteeing that Wikipedia is up to date with new concept pages and improving old concept pages over time. Using Wikipedia as the background knowledge is semi-dynamic in the sense that Wikipedia is continuously growing and we can easily use the latest Wikipedia version into our classification framework.
- Wikipedia provides a complex system of redirect and disambiguation pages, which could be leveraged to overcome the problems of limited coverage and lack of surface matching.
- Each content page in Wikipedia contains, in addition to the concept description, also semantic categories of the concept. We take advantage of both the text and the categories in supporting taxonomic relation classification.

Our algorithmic approach takes two well-segmented terms as input and outputs the predicted taxonomic relation between them, focusing on *ancestor-descendant* and *sibling* relations. We first exploit the Wikipedia structure to build semantic representation of each input term. Next, learning features are extracted from the semantic representations of the terms. A learned multi-class classifier is then applied to the extracted features to predict a probability distribution over the relations. In addition, we present an inference model that makes use of relational constraints and the aforementioned probability distribution over the taxonomic relations of the two input terms and *additional related terms* to enforce a coherent structure of terms and predicted relations that support the final taxonomic relation prediction.

In the rest of this paper, we present the overview of our algorithmic approach in Section 2. The learning component and the inference model of our approach are

Table 1. *4 taxonomic relations and some examples of each relation. Note that* London *is an ambiguous concept. It can be a city, thus a sibling of* Paris*, but can also refer to* Jack London*, thus a sibling of* Hemingway

.

		Examples	
Label	**Relation**	**Term** x	**Term** y
$x \leftarrow y$	x is an **ancestor** of y	actor	Mel Gibson
		food	rice
		wine	Champagne
$x \rightarrow y$	x is a **descendant** of y	Makalu	mountain
		Monopoly	game
		krooni	currency
$x \leftrightarrow y$	x and y are **siblings**	Paris	London
		copper	oxygen
		London	Hemingway
$x \not\leftrightarrow y$	x and y have **no relation**	Roja	C++
		egg	Vega
		HotBot	autism

described in Sections 3 and 4. Experimental results showing the advantages of our system are described in Section 5. We briefly discuss related work in Section 6, and conclude the paper in Section 7.

2 Algorithmic Approach

2.1 Preliminaries

The basic problem that we address in this work is the identification of fundamental taxonomic relations between any two well-segmented terms. Instead of building structured resources that record taxonomic relations among concepts as in previous work, our system focuses on directly classifying any two input terms into fundamental taxonomic relations including *ancestor-descendant*, *siblings* or *no relation.*

The main component of our system is a taxonomic relation classifier that is trained on supervised data consisting of pairs of terms and their taxonomic relations. In order to directly identify the directionality of the relations between input terms, we explicitly train and evaluate the classifier on four relation labels — *ancestor*, *descendant*, *sibling* and *no relation* . Some examples in the training data which consists of pairs of terms with four labels are shown in Table 1.

It is worth noting that it is a pragmatic decision to determine whether two terms hold a taxonomic relation. For example, according to the Wikipedia category

```
TAREC (Training)
INPUT:  Supervised data $\mathcal{D} = \{(x, y, rel)\}$
        Wikipedia $\mathcal{W}$
ALGORITHM:
    $\mathcal{D}' = \emptyset$
    For each $(x, y, rel) \in \mathcal{D}$
        $\mathfrak{R}_x \leftarrow$ WikiRepresentation$(x, \mathcal{W})$
        $\mathfrak{R}_y \leftarrow$ WikiRepresentation$(y, \mathcal{W})$
        $\mathcal{D}' = \mathcal{D}' \cup (\mathfrak{R}_x, \mathfrak{R}_y, rel)$
    $\mathcal{C} \leftarrow$ ExtractFeaturesAndTrainClasifier$(\mathcal{D}')$
RETURN: $\mathcal{C}$
```

Fig. 1. The TAREC training algorithm.

system, *George W. Bush* is a descendant of *Presidents of the United States* and also a descendant of *people*, *mammals*, and *organisms*. Without any constraint, the term *George W. Bush*, therefore, could be considered as a sibling of the term *oak (tree)* because they share *organisms* as a common subsumer. Obviously, we do not want to predict that *George W. Bush* and *oak* are siblings. In this work, we make use of Wikipedia structure as a source of background knowledge and use it to infer taxonomic relations between terms. Our taxonomic relation identifier uses a constant K – the maximum level to recursively climb up the Wikipedia category structure from a given concept – as a way to control determining of taxonomic relations between terms. Note that K is fixed for all relations and concepts

2.2 Overview

In this section, we present the overview of our **TA**xonomic **RE**lation **C**lassification (**TAREC**) system. The system consists of a training and an evaluation algorithm. Briefly, the training algorithm learns from a supervised training data set a local classifier that is used evaluation time in a constraint-based inference model to make the final prediction. We describe the algorithms below.

2.2.1 TAREC Training Algorithm

The training algorithm of TAREC is shown in Fig. 1. The input to the algorithm includes supervised training data $\mathcal{D}$ and Wikipedia data $\mathcal{W}$. The training data consists of examples in the form of triples (x, y, rel), where x and y are two terms and rel is their taxonomic relation. The relation rel denotes the taxonomic relation from x to y. For example, triple (*newspaper*, *New York Times*, ←) denotes that *newspaper* is an ancestor of *New York Times*, while (*Canada*, *country*, →) denotes that *Canada* is a descendant of *country*. Wikipedia data $\mathcal{W}$ is a local database constructed to allow access to necessary information in Wikipedia. We will discuss this background knowledge source in more details in Section 3.

To identify taxonomic relations between two single terms, we first map the terms

```
TAREC (Evaluation)
INPUT:  A pair of terms $(x, y)$
        Wikipedia $\mathcal{W}$
        Taxonomic relation classifier $\mathcal{C}$
ALGORITHM:
    $\mathfrak{R}_x \leftarrow$ WikiRepresentation$(x, \mathcal{W})$
    $\mathfrak{R}_y \leftarrow$ WikiRepresentation$(y, \mathcal{W})$
    $\mathcal{P}_{x,y} \leftarrow$ Classify$(\mathfrak{R}_x, \mathfrak{R}_y, \mathcal{C})$
    $\mathcal{Z}_{x,y} \leftarrow$ ExtractRelatedTerms$(x, y)$
    $rel \leftarrow$ ConstraintBasedInference$(\mathcal{P}_{x,y}, \mathcal{Z}_{x,y}, \mathcal{C})$
RETURN: $rel$
```

Fig. 2. The TAREC evaluation algorithm.

to some informative representations from which we could extract useful features. The function WikiRepresentation(*term*, $\mathcal{W}$) constructs a Wikipedia-based semantic representation for the input *term*. A new learning example is formed from the Wikipedia representation of the two input terms and their gold taxonomic relation. The new data is then used to train a local multi-class classifier ($\mathcal{C}$) to predict relations. Note that beside predicting relations, the learned classifier can also predict relation directionality due to the fact that we explicitly have four relation labels in the training data — x is an ancestor of y, x is a descendant of y, x and y are siblings, and x and y have no relation. We consider the classifier returned from the TAREC training algorithm as a *local* classifier to distinguish it from the global inference process employed in the TAREC evaluation algorithm.

2.2.2 TAREC Evaluation Algorithm

Given two terms (x, y), we apply the TAREC evaluation algorithm to predict their taxonomic relation. The evaluation algorithm uses the local classifier $\mathcal{C}$ learned using the TAREC training algorithm to predict the probability distribution over four taxonomic relation labels of (x, y) with background knowledge source $\mathcal{W}$. As we do when training, the two input terms are first mapped to a Wikipedia-based representation. The representations of x and y are then classified by $\mathcal{C}$ to get the probability distribution, $\mathcal{P}_{x,y}$, over the relation classes. Following that, the predicted probability distribution is used in a relational constraint-based inference model that takes advantage of other related concepts, $\mathcal{Z}_{x,y}$, of (x, y) to enforce a final coherent prediction on the taxonomic relation between x and y. In the inference model, we present a novel approach that leverages related concepts of two input terms, extracted from an existing knowledge source, to form a coherent relational structure that supports an accurate global prediction of taxonomic relations between input terms. The TAREC evaluation algorithm is summarized in Fig. 2.

3 Learning Local Taxonomic Relation Classifier

The TAREC training algorithm focuses on learning to predict the probability distribution over the possible taxonomic relations between two terms. It is clear that two single terms do not provide informative features to predict a relation between them. Our key idea is that we first map input terms to a more expressive representation space that allows us to extract rich features. To accommodate this idea, we take advantage of the structure of Wikipedia pages to map input terms to corresponding pages in Wikipedia.

Conceptually, Wikipedia provides a category structure. Thus, it may help us directly read off the taxonomic relations between terms. However, since terms could be ambiguous, this could lead to uncertain situations when they are mapped to Wikipedia pages (e.g. the term *Ford* could be mapped to both *Ford Motor Company* and president *Gerald Ford.*) Furthermore, even if a term is mapped to a Wikipedia page correctly, it is not easy to directly use the Wikipedia category system to infer its relation to another term due to the fact that the taxonomic relation information may be hidden in the text of their Wikipedia pages, not simply in the categories. For example, *Bill Clinton* is a descendant of *American*, but there is no explicit Wikipedia category *American* in the Wikipedia page of *Bill Clinton*. Nevertheless, the fact that there are indeed categories *American health activists* and *American humanitarians* on the *Bill Clinton* Wikipedia page would be very helpful to inferring its taxonomic relation to the term *American*.

In this section, we first briefly describe the structure of Wikipedia pages, then we introduce two mapping procedures that produce different behaviors in our final systems, and finally, we present the learning features.

3.1 The Structure of Wikipedia Pages

The majority of Wikipedia pages provide information about concepts (or entities). Typically, each concept page consists of three important pieces of information: a title (usually identical to the concept surface form), a body text which describes the concept, and the categories to which the concept belongs. The upper part of Table 2 shows snippets of some regular pages exemplifying the information of concepts *President of the United States*, *George W. Bush* and *Gerald Ford*.

In addition, it is common for a concept to be referred to in multiple ways. For example, *Gerald Ford* can also be referred to as *Gerald R. Ford*, *Gerald Rudolph Ford, Jr.* or *President Ford*. Fixed resources, such as WordNet and the Extended WordNet, are not able to deal with this issue, whereas the Wikipedia page structure provides an excellent resource to address this problem. The reason is that Wikipedia maintains a huge system of redirect pages that redirects uncanonical concepts to their canonical form. The middle part of Table 2 illustrates some redirect pages and their references. Redirect pages usually do not have categories because the categories are maintained on corresponding canonical pages.

Furthermore, a term may be ambiguous and could refer to multiple concepts. Fortunately, Wikipedia provides a clear organization of ambiguous concepts in a

Table 2. *An excerpt of the structure of Wikipedia pages*

Page Title	**Text**	**Categories**
	Regular (Non-Redirection) pages	
President of the United States	The President of the United States is the head of state and head of government of the United States and is the highest political official in the United States by influence and recognition. The President leads the executive branch of the federal government and is one of only two elected members of the executive branch...	Presidents of the United States, Presidency of the United States
George W. Bush	George Walker Bush; born July 6, 1946) served as the 43rd President of the United States from 2001 to 2009. He was the 46th Governor of Texas from 1995 to 2000 before being sworn in as President on January 20, 2001...	Children of Presidents of the United States, Governors of Texas, Presidents of the United States, Texas Republicans...
Gerald Ford	Gerald Rudolff Ford (born Leslie Lynch King, Jr.) (July 14, 1913 December 26, 2006) was the 38th President of the United States, serving from 1974 to 1977, and the 40th Vice President of the United States serving from 1973 to 1974.	Presidents of the United States, Vice Presidents of the United States, Republican Party (United States) presidential nominees...
	Redirect pages	
US President	#Redirect [[President of the United States]]	(N/A)
Gerald R. Ford	#Redirect [[Gerald Ford]]	(N/A)
	Disambiguation pages	
Ford	#Refer [[Ford Motor Company]] #Refer [[Gerald R. Ford]] #Refer [[Henry Ford]] #Refer [[Ford's Theatre]]	Disambiguation page, Surnames
table	#Refer [[Table (furniture)]] #Refer [[Table (information)]] #Refer [[Table (database)]]	Disambiguation page

special page structure which consists of disambiguation pages. Each disambiguation page contains several concepts that an ambiguous term may refer to. The last part of Table 2 shows the disambiguation pages of two terms, *Ford* and *table*, and the concepts they refer to. Note that it is possible for a referred concept to be linked to a redirect page. For instance, *Ford* may refer to *Gerald R. Ford* which is, in turn, redirected to canonical the concept *Gerald Ford*.

Together, all these pieces of information make Wikipedia page structure a valuable resource when building a semantic representation for input terms.

3.2 Wikipedia-based Semantic Representaiton

In this section, we present two approaches to constructing Wikipedia-based semantic representations for input terms. Both approaches are motivated by the intuition that real-world applications are usually interested in identifying relation between related terms rather than arbitrary ones. For example, it is more likely that the term *Ford* in the pair (*George W. Bush*, *Ford*) refers to the former president of the United States, *Gerald Ford*, than to the car manufacturer *Ford Motor Company* or its founder *Henry Ford.* Our approaches below take this intuition into account when constructing the term's Wikipedia semantic representation.

In the following section, we use *Wikipedia concept* and *Wikipedia page* interchangeably to refer to the Wikipedia page associated with a concept expressed by the page title. For example, the Wikipedia page *Gerald Ford* is associated with the Wikipedia concept *Gerald Ford.*

3.2.1 Matching-based Approach

Intuitively, given a term, this approach looks for the most appropriate Wikipedia page that best matches (i.e. best describes) the term. To this end, the matching-based approach maps the term to Wikipedia pages by directly looking it up and matching it with Wikipedia pages' title. Beside regular pages, we make use of both redirect and disambiguation pages. Given a pair of terms, the output of this procedure includes two Wikipedia pages that provide the best description of the two input terms, respectively.

In this approach, each Wikipedia concept page, p_x (where x is a Wikipedia concept), is represented by a set of keywords, KW_x. The keywords are extracted by selecting the top tokens in the body text and the categories of the page ranked by their TF-IDF scores. In this work, for each Wikipedia page, we use the first paragraph of the body text as an approximation for the whole text. We use the Porter stemmer to normalize the tokens.[3] For example, Wikipedia page *Gerald Ford* is represented by the following list of normalized tokens {*ford, presid, amend, gerald, vice, fifth, serv, fortieth, state, rudolph, nixon, unit, resign, eighth, thirti, constitut, twenti, term, person, bachil, episcopalian, adopte, waterg, wolverin, cardiovascular, nomine, recipi, communist, lawyer, rapid, omaha, scout, death, descend, yale, alumni, eagl*}. In our experiments, we used a maximum of top 40 keywords of KW_x, including the top 20 keywords of the text and the top 20 keywords from the categories of x.

Furthermore, each Wikipedia concept x is characterized by an absolute prominence score, α_x, which is defined as the number of times it is hyperlinked in the whole Wikipedia corpus. Intuitively, the prominence notion of a concept encodes its popularity by measuring how often it is linked to from other Wikipedia pages. Given a pair of unambiguous concepts (x, y), we define their similarity as follows:

[3] http://tartarus.org/∼martin/PorterStemmer/

$$sim(x, y) = \alpha_x \times \alpha_y \times |KW_x \cap KW_y|$$

For a disambiguation page DP_x of term x (e.g. $x =$ *Ford* as shown in Table 2), each referred concept $u \in DP_x$ is assigned a relative prominence score $\alpha_u^x = \frac{\alpha_u}{max_{u' \in DP_x} \alpha_{u'}}$, where α_u is the absolute prominence scores of u. Given a concept $u \in DP_x$ and a concept $v \in DP_y$, we define the similarity score of pair (u, v) as follows: $sim(u, v) = \alpha_u^x \times \alpha_v^y \times |KW_u \cap KW_v|$. In general, if x is unambiguous (i.e. x matches a normal page or a redirected page in Wikipedia), its absolute prominence score is used. Otherwise, relative prominence score is used in the similarity metric.

Let $\mathcal{W}_{DP}$ be the list of Wikipedia disambiguation pages, $\mathcal{W}_R$ be the list of redirect pages, and $\mathcal{W}_{NR}$ be the list of regular (non-redirection) pages. We use $\mathcal{W}_{DP}(x)$, $\mathcal{W}_R(x)$ and $\mathcal{W}_{NR}(x)$ to denote the functions that map term x to the best corresponding Wikipedia page in $\mathcal{W}_{DP}$, $\mathcal{W}_R$ and $\mathcal{W}_{NR}$, respectively. A term is mapped to a Wikipedia page via an exact string matching operation between the term and the title of the page.

Given input pair (x, y), the matching-based approach follows the procedure sketched below to select the best Wikipedia page for each input term.

```
 1. Input: A pair of well-segmented terms $(x, y)$.
 2. $Pool_x = \emptyset$; $Pool_y = \emptyset$
 3. if $x$ in $\mathcal{W}_{DP}$ // $x$ is an ambiguous concept
 4.         $DP_x = \mathcal{W}_{DP}(x)$
 5.         $Pool_x \leftarrow$ {the concepts in $DP_x$}
 6. else if $x$ in $\mathcal{W}_R$ // $x$ is an unambiguous concept, but redirected
 7.         $R_x = \mathcal{W}_R(x)$
 8.         $Pool_x \leftarrow$ {the redirected concept in $R_x$}
 9. else if $x$ in $\mathcal{W}_{NR}$ // $x$ is an unambiguous (non-redirection) concept
10.         $NR_x = \mathcal{W}_{NR}(x)$
11.         $Pool_x \leftarrow \{NR_x\}$
12. Similarly, extract $Pool_y$ for $y$ as from step 2 to 10.
13. Find the best pair of pages $(u^*, v^*) = argmax_{u \in Pool_x, v \in Pool_y} sim(u, v)$
14. Return: $\mathfrak{R}_x = \{u^*\}$ and $\mathfrak{R}_y = \{v^*\}$.
```

Note that if x is unambiguous, x is mapped to a single Wikipedia page, and $Pool_x$, therefore, has only one single member. In this case, the absolute prominence score α_x is used in the similarity scoring function. This is similar for y and $Pool_y$.

3.2.2 Search-based Approach

The key idea in our second approach is that we look for a set of relevant pages in the Wikipedia corpus to be used as a representation of a term, rather than a single page as in the matching-based approach. This approach requires information retrieval techniques to search and retrieve relevant Wikipedia pages. In this work, we use the local search engine Lucene.[4] The main procedure of this approach proceeds as follows:

1. Input: A pair of well-segmented terms (x, y).
2. Create a unified query by concatenating *x AND y*. For example, for pair (*George W. Bush*, *Gerald Ford*), the unified query is *George W. Bush AND Gerald Ford.*
3. Search the complete Wikipedia corpus text using the unified query to retrieve a list of relevant pages, $\mathcal{L}_{x,y}$.
4. Extract the top important keywords from the categories of the pages in $\mathcal{L}_{x,y}$ by ranking them using TF-IDF scores. Intuitively, this search will retrieve relevant pages for both input terms, so the top extracted keywords will tie the semantic meaning of the two input terms to each other. For example, the unified query in step (1) will retrieve relevant pages of both *George W. Bush* and *Gerald Ford.* From the retrieved pages, extracted keywords may include: *president*, *politician*, *united*, *state*, etc..
5. Concatenate each input term with the list of keywords extracted in step 4. For instance, *George Bush* will be augmented to make a conjunctive query: *George W. Bush* AND *president* AND *politician* AND *united* AND *state.*
6. Search for the top relevant pages, $\mathfrak{R}_x$ and $\mathfrak{R}_y$ of x and y using their new queries from step 5.
7. Return: $\mathfrak{R}_x$ and $\mathfrak{R}_y$ as the Wikipedia representations of x and y, respectively.

In our experiments, we use the top 10 keywords in step 4, and 10 Wikipedia pages as the maximum number of pages in the Wikipedia representation of each term returned in step 7.

3.3 Feature Extraction

The features of a pair of terms are extracted from their Wikipedia representations. As discussed earlier (Section 3.1), a regular Wikipedia page of a Wikipedia concept usually consists of a title, a body text, and a list of categories to which the concept belongs. For convenience, for a term x, we use *the titles of* x, *the text of* x, and *the categories of* x to refer to the titles, text, and categories of the associated pages in the representation of x. Table 3 shows a short version of the Wikipedia representation of two input terms *Gerald Ford* and *Bush* extracted by the search-based approach. Note that the pages in the Wikipedia representation of *Bush* are mostly about presidency because the term is influenced by the other term *Gerald Ford*, as expected in the search-based approach. In this context, *the titles of Gerald Ford*, *the text of Gerald Ford* and *the categories of Gerald Ford* consist of all the titles, the body text and the categories of the Wikipedia pages in the Wikipedia representation of *Gerald Ford*, respectively. Similar notions apply to the term *Bush.*

In addition to the direct categories of a Wikipedia page of a term, we also collect

[4] E.g. http://lucene.apache.org/

Table 3. *A short version of the Wikipedia representation of input pair* (Bush, Gerald Ford)*. Note that page* Presidency of Gerald Ford *is redirected to page* Gerald Ford*; they, therefore, get the same text and category list.*

Term	**Page Title**	**Text**	**Category**
Gerald Ford	Gerald Ford	Gerald Rudolff Ford (born Leslie Lynch King, Jr.) (July 14, 1913 December 26, 2006) was the 38th President of the United States, serving from 1974 to 1977, and the 40th Vice President of the United States serving from 1973 to 1974...	Presidents of the United States, Vice Presidents of the United States, Republican Party (United States) presidential nominees...
	Presidency of Gerald Ford	Gerald Rudolff Ford (born Leslie Lynch King, Jr.) (July 14, 1913 December 26, 2006) was the 38th President of the United States, serving from 1974 to 1977, and the 40th Vice President of the United States serving from 1973 to 1974...	Presidents of the United States, Vice Presidents of the United States, Republican Party (United States) presidential nominees...
	Electoral history of Gerald Ford	Electoral history of Gerald Ford, 38th President of the United States and 40th Vice President of the United States...	Gerald Ford, Electoral history of American politicians...
Bush	George W. Bush	George Walker Bush; born July 6, 1946) served as the 43rd President of the United States from 2001 to 2009. He was the 46th Governor of Texas from 1995 to 2000 before being sworn in as President on January 20, 2001...	Children of Presidents of the United States, Governors of Texas, Presidents of the United States, Texas Republicans...
	George H. W. Bush	George Herbert Walker Bush (born June 12, 1924) is an American politician who served as the 41st President of the United States (198993)...	Parents of Presidents of the United States, Presidents of the United States, Texas Republicans...
	Presidency of George W. Bush	The presidency of George W. Bush began on January 20, 2001, when he was inaugurated as the 43rd President of the United States of America...	Presidencies of the United States, Presidency of George W. Bush...

its higher level categories: we start from the categories of the page in its representation and recursively go up K levels on the Wikipedia category system as before. The categories of a term are the union of its direct categories and all the categories of the upper level pages.

Below we present the features extracted for an input pair of terms, (x, y), that will be used in learning the relations classifier. All features are real value.

Bag-of-words Similarity: We define four bag-of-words features as the degree of similarity among the texts and categories of x and y. The features are shown in Table 4. We use the cosine similarity metric to measure the value of these features.

Table 4. *Bag-of-word similarity features of (x,y), where texts(*term*) and categories(*term*) are the functions that extract associated texts and categories from the semantic representation of* term*.*

Bag-of-words similarity features
text(x) vs. categories(y)
categories(x) vs. text(y)
text(x) vs. text(y)
categories(x) vs. categories(y)

Let $v_x^t =< w_1, w_2, \cdots >$ be the bag-of-words feature vector of the texts of term x, where w_i is the indicator variable that indicates whether a particular word at position i is present in the text of x. Let $v_y^c =< w_1, w_2, \cdots >$ be the bag-of-words feature vector of the category of y. The similarity strength between the text of x and the categories of y is measured as in Equation (1).

$$sim(v_x^t, v_y^c) = \frac{\overrightarrow{v_x^t} \bullet \overrightarrow{v_y^c}}{\|\overrightarrow{v_x^t}\| \|\overrightarrow{v_y^c}\|} \tag{1}$$

For the other three bag-of-words features, we use similar notions and formulas.

Association Information: This feature measures the association information between terms by considering their information overlap over the whole Wikipedia data. We capture this feature using pointwise mutual information (*PMI*) which quantifies the discrepancy between the probability of two terms appearing together versus the probability of each term appearing independently.[5] The PMI of two terms x and y is estimated as in Equation (2):

$$PMI(x,y) = log\frac{p(x,y)}{p(x)p(y)} = log\frac{Nf(x,y)}{f(x)f(y)} \tag{2}$$

where N is the total number of Wikipedia pages, and f is a counting function that returns the number of times its argument(s) appear(s) (together) in Wikipedia.

Overlap Ratios: The overlap ratio features capture the fact that the titles of an ancestor term usually overlap with the categories of its descendants. Similarly, the categories of two sibling terms are also usually highly overlapping. For example, Wikipedia page *Presidents of the United States*, as shown in Table 2, has a title that overlaps with one of the categories of the Wikipedia page *George W. Bush.* This evidence strongly supports the conclusion that the term *Presidents of the United*

[5] *PMI* is different than mutual information. The former applies to specific outcomes, while the latter is used to measure the mutual dependence of two random variables.

Table 5. *Overlap ratio features of (x,y), where titles(*term*) is a function that returns the titles of the Wikipedia pages in the Wikipedia representation of* term*; function categories(*term*) was defined in Table 4*

Overlap ratio features
titles(x) vs. categories(y)
categories(x) vs. titles(y)
categories(x) vs. categories(y)

States is an ancestor of the term *George W. Bush*. On the other hand, the categories of *George W. Bush* and *Gerald Ford* overlap with each other in several categories, such as *Presidents of the United States*. In general, a higher overlap ratio indicates a better chance for two terms to hold a taxonomic relation. We use three overlap ratio features as shown in Table 5.

We measure the overlap ratios by the ratios of the numbers of *key phrases* in the titles and categories of the input terms. In our context, a phrase is considered to be a key phrase if it belongs to one of the following types:

- the whole string of a title or category
- the lemma of the head a category
- the post-modifier of a category

We use the Noun Group Parser (Suchanek *et al.*, 2007) to extract the head and post-modifier of a category. For example, the category *Cities in Illinois* of Wikipedia page *Chicago* could be parsed into a head in its root form, *City*, and a post-modifier, *Illinois*. Therefore, in the pair of terms *(City, Chicago)*, the term *City* overlaps with the head, *City*, of the category *Cities in Illinois* of the Wikipedia page *Chicago*. This is a strong feature indicating that *Chicago* is a descendant of *City*.

Let two input terms be x and y. Let $u_x^t = (t_x^1, t_x^2, \cdots)$ denote the set of titles of term x in its Wikipedia representation. Also, let $u_y^c = (c_y^1, c_y^2, \cdots)$ be the set of the key phrases of the categories of term y in its Wikipedia representation. The overlap ratio feature between the titles of term x and the categories of term y is computed using the Jaccard similarity coefficient metric as shown in Equation (3).

$$overlap(x, y) = \frac{|u_x^t \bigcap u_y^c|}{|u_x^t \bigcup u_y^c|} \tag{3}$$

For the other two overlap ratio features, we use similar notions and formulas. In addition, to measure the overlap ratio feature between the categories of the two input terms, the post-modifiers of the categories are not used because when the categories of the terms are compared together, the overlap of the post-modifiers of the categories is not useful (e.g. categories *Actors of America* and *Companies of America* overlap in their post-modifiers *America*, but this overlap does not help to recognize taxonomic relations).

Overall, we use eight feature types for the local classifier including: bag-of-words features (4), association information (1), and overlap ratio features (3).

3.4 Non-Wikipedia Terms

Although most commonly used terms have corresponding Wikipedia pages, new entities and concepts always come up and there are still many terms that do not have Wikipedia pages. We call these terms *non-Wikipedia terms*. In order to handle these terms, we propose to use a normalization procedure to find approximate Wikipedia pages for non-Wikipedia terms. The basic idea of the normalization procedure is to find a replacement for a non-Wikipedia term which, ideally, keeps the underlying taxonomic relation unchanged, by using Web search. For example, given input pair (*Lojze Kovačič*, *Rudi Šeligo*), there is no English Wikipedia page for *Lojze Kovačič*, who is a writer, but if we can find another writer, such as *Marjan Rožanc*, and use it as a replacement of *Lojze Kovačič*, then we can continue classifying the taxonomic relation of pair (*Marjan Rožanc*, *Rudi Šeligo*).

Our Wikipedia normalization procedure follows (Sarmento *et al.*, 2007). We first compose a query concatenating the two input terms (e.g. *Lojze Kovačič AND Rudi Šeligo*) and use Web search[6] to retrieve list-structure snippets with the following pattern: "... $\langle del \rangle$ c_a $\langle del \rangle$ c_b $\langle del \rangle$ c_c $\langle del \rangle$ ..." (the two input terms must be among c_a, c_b, c_c, ...). In the pattern, *del* is a delimiter and could be commas, periods, or asterisks.[7] Using the snippets that contain the patterns of interest, we extract c_a, c_b, c_c etc. as replacement candidates. To reduce noise, we empirically constrain the list to contain at least 4 terms that are no longer than 20 characters each.[8] The candidates are ranked based on their occurrence frequency. The top candidate for which we can construct a Wikipedia representation, is used as a replacement.

4 Global Inference with Relational Constraints

In this section, we present a novel inference model which relies on the structure of pair-wise mutual taxonomic relations among two input terms and some additional related terms to enforce final coherent prediction. The main idea of our inference model is that logical constraints on relations among terms may prevent predicting illegitimate structures. Our global objective, therefore, focuses on selecting the best taxonomic relation between two input terms that allows legitimate structures to be formed when additional terms are taken into account. For example, given two target terms *George W. Bush* and *president*, we add an additional related term, such as *Bill Clinton*; if we can identify, with some degree of confidence, that (i) *president* is an ancestor of *Bill Clinton*, and (ii) *Bill Clinton* is a sibling of *George W. Bush*, then due to the transitivity property of taxonomic relations, the term *George W.*

[6] http://developer.yahoo.com/search/web/
[7] Periods and asterisks capture enumerations.
[8] We believe that a list with less than 4 terms may not be a good list. Furthermore, we require that a candidate term has no more than 20 characters to prevent noisy terms.

Bush is likely to be a descendant of the term *president* since other relations will create illegitimate structures.

The two input terms along with some additional related terms and the taxonomic relations among them form a structure that we call a *term network* (or *network* for short). Fig. 3 shows some n-term networks consisting of two input terms (x, y), and additional terms v, w, z. Note that the arrows in the figures follow the notions in Table 1.

The aforementioned observations suggest that if we can get additional terms that are related to the two input terms, we can enforce coherent structures and eliminate illegitimate combinations of terms and relations via relational constraints. This would help the system improve the predictions of taxonomic relations of input pairs. In this work, we formalize our inference model using constraint-based formulations that were introduced to the NLP community in (Roth and Yih, 2004) and were shown to be very effective in exploiting declarative background knowledge (Denis and Baldridge, 2007; Punyakanok *et al.*, 2008; Chang *et al.*, 2008).

4.1 Enforcing Relational Constraints through Global Inference

The main goal of our inference model is to eliminate illegitimate term networks and select the best taxonomic relation of two input terms, embedded in legitimate structures. Below, we formalize our inference model with the following notation:

- (x, y) : two input terms.
- $\mathcal{Z}_{x,y} = \{z_1, z_2, ..., z_m\}$: a set of additional terms.
- $Z \subseteq \mathcal{Z}_{x,y}$: a subset of terms in $\mathcal{Z}_{x,y}$.
- e : an edge imposing a relation between two terms; e can be one of four relations.
- $w(e)$: the weight of e, given by local classifier $\mathcal{C}$ (see Fig. 1). Recall that $\mathcal{C}$ predicts a probability distribution over four taxonomic relations between x and y.

Each network is formed by x, y and the terms in Z. Let $l = |Z|$, then there are $n = 2 + l$ terms in each network, and $4^{\left[\frac{1}{2}n(n-1)\right]}$ networks can be constructed.

We define a *relational constraint* as a network that imposes an *illegitimate structure* on its edges. That is, a constraint is *unlexicalized* in the sense that we only consider the edge structure of the network, regardless of the specific terms. In this work, we focus on 3-term networks (i.e. $l = 1$). For example, given input pair (*red*, *green*) and $\mathcal{Z} = \{blue, yellow\}$, we can construct 64 networks for triple $\langle red, green, Z = \{blue\}\rangle$ and 64 networks for $\langle red, green, Z = \{yellow\}\rangle$ by trying all possible relations between the terms.

Fig. 3(c) shows a relational constraint where the term *red* is a sibling of both *green* and *blue*, but *green* is an ancestor of *blue*; this structure is illegitimate because of the transitivity property. The relational constraints in this work are manually constructed. In the case of 3-term networks, constraints are written in a clockwise direction, starting from the two input terms, (x, y). For instance, the illegitimate

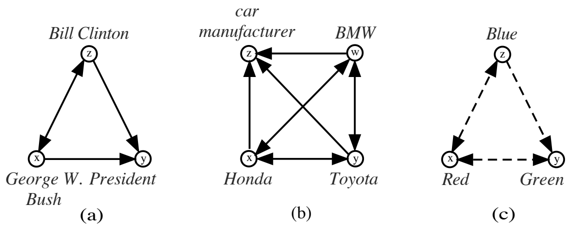

Fig. 3. Examples of n-term networks with input pair (x, y). (a) and (b) show two valid structures, whereas (c) illustrates a relational constraints with an illegitimate structure.

structure in Fig. 3(c) forms the following relational constraint: $\langle \leftrightarrow, \leftrightarrow, \rightarrow \rangle$, where the arrows follow the notation in Table 1.

We solve this constraint optimization problem by a 2-stage greedy approach which is integrated into the Constrained Conditional Model (CCM) (Chang *et al.*, 2008). We first check and eliminate all term networks that are illegitimate, then greedily select the best taxonomic relation that allows legitimate networks.[9]

Let $\mathcal{RC}$ be a list of relational constraints. Network t can be assigned a score using the network scoring function defined in Eq. (4). This scoring function is a linear combination of the edge weights, $w(e)$, of the edges in t and the penalties, ρ_k, that penalize if the edge structure of t belongs to $\mathcal{RC}$.

$$score(t) = \sum_{e \in t} w(e) - \sum_{k=1}^{|\mathcal{RC}|} \rho_k d_{\mathcal{RC}_k}(t) \tag{4}$$

where, function $d_{\mathcal{RC}_k}(t)$ indicates whether t matches $\mathcal{RC}_k$.

We can define relational constraints as either hard or soft constraints. In the current work, we consider illegitimate networks as hard constraints: all term networks that belong to the list of relational constraints are simply discarded. To do this, we set penalty factor ρ_k to ∞, for all $\mathcal{RC}_k$.

Now, among the set of networks formed by $\langle x, y, Z \rangle$, we select the best network, $t^*_Z = \text{argmax}_t score(t)$.

Let $t^*_\cap = \cap_Z t^*_Z$, $Z \subseteq \mathcal{Z}$; we then partition $t^*_\cap$ into four groups according to the relation, denoted by rel, between x and y in each network. Let us denote each group by $\mathcal{T}_{rel}$. To choose the best taxonomic relation, rel^*, between x and y, we pick the relation which maximizes the average score of the whole group as in Eq. (5).

$$rel^* = \text{argmax}_{rel} \frac{1}{|\mathcal{T}_{rel}|} \sum_{t^* \in \mathcal{T}_{rel}} \lambda_{t^*} score(t^*) \tag{5}$$

where λ_t is the weight of the unlexicalized term network t, defined as the occurrence probability of t in an augmented version of the training data. To augment the

[9] We do not use an exact inference approach (e.g. Integer Linear Programming (ILP)) to solve the problem because the optimization problem here with 3-term networks is small and can be effectively solved by a greedy approach. However, ILP and other optimization approaches could be used as the alternatives to our greedy approach.

YAGO Query Patterns

INPUT: term x

OUTPUT: lists of ancestors, siblings, and children of x

Pattern 1	Pattern 2	Pattern 3
x MEANS ?A	x MEANS ?A	x MEANS ?D
?A SUBCLASSOF ?B	?A TYPE ?B	?E TYPE ?D
?C SUBCLASSOF ?B	?C TYPE ?B	

RETURN: ?B, ?C, ?E as
lists of ancestors, siblings (extracted by Patterns 1 and 2),
and children (extracted by Pattern 3), respectively.

Fig. 4. Our YAGO query patterns used to obtain related terms for x.

training data, we first extract additional terms for each pair of terms in the training data, and then apply our local classifier to identify the taxonomic relation between the terms. The weight of a network t is computed as the number of time t occurs in the augmented training data divides by the total number of term networks.

4.2 Extracting Related Terms

In the inference model, we need to obtain additional terms, $\mathcal{Z}_{x,y}$, that are related to x and y. Hereafter, we refer to additional terms as *related terms*. The related term space is composed of the direct ancestors, siblings and direct children of the input terms, obtained from some knowledge source.

We propose to extract related terms from the YAGO ontology (Suchanek *et al.*, 2007). YAGO is chosen over the Wikipedia category system used in our work because YAGO is a clean ontology built by carefully combining Wikipedia and WordNet.[10]

In the YAGO model, all objects (e.g. *cities*, *people*, etc.) are represented as *entities*. To map our input terms to entities in YAGO, we use the MEANS relation defined in the YAGO ontology. Furthermore, similar entities are grouped into *classes*. This allows us to obtain direct ancestors of an entity by using the TYPE relation which gives the entity's classes. Furthermore, we can get ancestors of a class with the SUBCLASSOF relation.[11] By using three relations, MEANS, TYPE and SUBCLASSOF, in the YAGO model, we can obtain direct ancestors, siblings, and direct children, if any, for input terms. In the case that the two input terms are not contained in YAGO, the inference model is simply ignored. Fig. 4 presents three patterns that

[10] However, YAGO by itself is weaker than our system in identifying taxonomic relations (see Section 5).

[11] These relations are defined in the YAGO ontology.

we use to query related terms from YAGO. For more details on YAGO relations, we refer readers to (Suchanek *et al.*, 2007).

5 Experimental Study

In this section, we evaluate TAREC against other systems built upon existing well-known knowledge sources. The resources are either hierarchical structures or extracted by using distributional semantic models. We also provide experimental analyses on the compared systems.

5.1 Comparison to Hierarchical Structures

5.1.1 Data Preparation

We create and use two main data sets in these experiments.

Dataset-I is generated from 40 semantic classes of about 11,000 instances. The original semantic classes and instances were manually constructed with a limited amount of manual post-filtering and were used to evaluate information extraction tasks in (Paşca, 2007; Paşca and Van Durme, 2008) (we denote this original data as **OrgData-I**). This data set contains both terms with Wikipedia pages (e.g. *George W. Bush*) and non-Wikipedia terms (e.g. *hindu mysticism*). Pairs of terms are generated by randomly pairing semantic class names and instances. We generate disjoint training and test sets of 8,000 and 12,000 pairs of terms, respectively. We call the test set of this data set **Test-I**.

Dataset-II is generated from 44 semantic classes of more than 10,000 instances used in (Vyas and Pantel, 2009).[12] The original semantic classes and instances were extracted from Wikipedia lists. This data therefore contains only terms that have Wikipedia pages. We also generate disjoint training and test sets of 8,000 and 12,000 pairs of terms, respectively, and call the test set of this data set **Test-II**.

Both data sets contain both types of closed semantic classes (e.g. *chemical element*, *country*) and open semantic classes (e.g. *basic food*, *hurricane*). Moreover, there are classes with proper nouns (e.g. *actor* with *Mel Gibson*) and classes with common nouns (e.g. *basic food* with *rice*, *milk*).

Many semantic class names in the original data sets are written in short forms. We expand these names to meaningful names that are used by all systems in our experiments. For example, *terroristgroup* is expanded to *terrorist group*, *terrorism*, *chemicalelem* to *chemical element*, *proglanguage* to *programing language*. Some examples are shown in Table 1. Four types of taxonomic relations are covered with balanced number of examples in all data sets.[13]

To evaluate our systems, we used a snapshot of Wikipedia from July, 2008. After cleaning and removing articles without categories (except redirect pages), 5,503,763

[12] There were 50 semantic classes in the original data set. We grouped some semantically similar classes for the purpose of classifying taxonomic relations.

[13] Published at http://cogcomp.cs.illinois.edu/page/resources/TaxonomicRelationData.

articles remained. We indexed these articles by their body texts using Lucene.[14] In practice, we only indexed the abstract (usually the first paragraph) of the Wikipedia pages. All characters were lower-cased and all punctuations were removed. We also removed stop words.[15] Furthermore, when performing search on the Wikipedia index, we did not normalize the search similarity score to the length of an article. Specifically, we overwrote the *lengthNorm* function in Lucene to always return value 1. All query tokens must occur in a Wikipedia page for it to be returned in the search result list.

As a learning algorithm, we used the Regularized Averaged Perceptron (Freund and Schapire, 1999) within the LBJ modeling language (Rizzolo and Roth, 2010).[16] The learning algorithm used the one-vs-all scheme to transform a set of binary classifiers into a multi-class classifier. The raw activation scores were converted into probability distribution with the *softmax* function (Bishop, 1996). If there are n classes and the raw score of class i is act_i, the posterior estimation for class i is:

$$Prob(i) = \frac{e^{act_i}}{\sum_{1 \leq j \leq n} e^{act_i}}$$

5.1.2 Compared Systems

Beside TAREC, we developed three other systems built upon well-known large-scale hierarchical structures.

Strube07 is built on the latest version of a taxonomy, T_{Strube}, which was derived from Wikipedia (Ponzetto and Strube, 2007). It is worth noting that the structure of T_{Strube} is similar to the page structure of Wikipedia. For a fair comparison, we first generate a Wikipedia representation for each input term by following search-based approach in Section 3.2.2. The titles and categories of the articles in the representation of each input term are then extracted. Only titles and their corresponding categories that are in T_{Strube} are considered. A term is an ancestor of another one if at least one of its titles is in the categories of the other term. If two terms share a common category, they are considered siblings, otherwise they are considered to have no relation. The ancestor relation is checked first, then the sibling, and finally no relation.

Snow06 uses the Extended WordNet (Snow *et al.*, 2006). Words in the Extended WordNet can be common nouns or proper nouns. Given two input terms, we first map them onto the hierarchical structure of the extended WordNet by exact string matching. A term is an ancestor of another one if it can be found as a subsumer after recursively going up K levels in the hierarchical tree of the Extended WordNet from the other term. If two terms share a common subsumer within K levels on the

[14] http://lucene.apache.org, version 2.3.2

[15] We used the following stop word list: *a, about, an, are, as, at, be, by, com, de, en, for, from, how, i, in, is, it, la, of, on, or, that, the, this, to, was, what, when, where, who, will, with, und, the, www.*

[16] http://cogcomp.cs.illinois.edu/page/software_view/11

Table 6. *Performance, in accuracy, of the systems on* **Test-I** *and* **Test-II**. *TAREC systems with local models simply use the local classifier to classify taxonomic relations by choosing the relation having highest confidence.*

System	Test-I	Test-II
Strube07	24.32	25.63
Snow06	41.97	36.26
Yago07	65.93	70.63
Local		
TAREC^{MATCH}	79.64	77.56
TAREC^{SEARCH}	81.89	84.7
Inference		
TAREC^{SEARCH}	**85.34**	**86.98**

tree, they are classified as siblings. Otherwise, there is no relation between them. Similar to Strube07, we first check ancestor, then sibling, and finally no relation.

Yago07 uses the YAGO ontology (Suchanek *et al.*, 2007) as its main source of background knowledge. Because the YAGO ontology is a combination of Wikipedia and WordNet, this system is expected to perform well in identifying taxonomic relations. To access term's ancestors and siblings, we use patterns 1 and 2 in Fig. 4 to map a term to the ontology and move up on the ontology. The relation identification process is then similar to those of Snow06 and Strube07. If an input term is not recognized by these systems, they are considered to have no relation.

Our TAREC evaluation algorithm is described in Fig. 2 and is evaluated in two settings: **TAREC**^{MATCH}, which employs the matching-based approach (Section 3.2.1), and **TAREC**^{SEARCH}, which uses the search-based approach (Section 3.2.2).

We evaluate each setting with the **Local** model which does classification on term pairs by directly selecting the highest-probability relation returned by the local classifier $\mathcal{C}$. For the **Inference** model, we manually construct a pre-defined list of 35 relational constraints.

5.1.3 Results

In all systems compared, we vary the value of K from 1 to 4. The best results of the systems are reported.

Table 6 shows the comparison of all systems evaluated on both Test-I and Test-II. Our TAREC (Local) systems, as shown, significantly outperform the other systems. The results show that our machine learning-based classifier is very flexible in extracting features of the two input terms and is thus much better at predicting their taxonomic relation. In contrast, because other systems rely heavily on string matching techniques to map input terms to their respective ontologies, they are very inflexible and brittle. This clearly shows the limitations of using structured resources to classify taxonomic relations.

Between the local systems, the search-based approach is better than the matching-based approach. This can be explained by the fact that the matching-based approach is still not flexible enough in mapping input terms to Wikipedia representation.

We apply the inference model on top of TAREC^{SEARCH} (Local) and further achieve remarkable improvement. The improvement of TAREC^{SEARCH} (Inference) over TAREC^{SEARCH} (local) on Test-I shows the contribution of both the normalization procedure (see Section 3.4) and the global inference model to the classification decisions, whereas the improvement on Test-II emphasizes only the contribution of the inference model, because Test-II only contains terms that have corresponding Wikipedia pages. This improvement also suggests that relational constraints help improve the local classifier by enforcing coherent decisions over underlying structures of terms and relations.

Furthermore, it is also interesting to see that between Test-I and Test-II test sets, TAREC^{MATCH} (Local) performs better on Test-I. Our analysis shows that this is because there are more ambiguous terms (i.e. requiring more mappings to concepts in disambiguation pages) in Test-II than Test-I, therefore, Test-II is more difficult than Test-I for the matching-based approach. Specifically, 36.23% of terms in Test-II are ambiguous, while that number in Test-I is 31.71%.

For the value of K, the best results of the systems on Test-I are achieved with: $K = 4$ for Strube07, $K = 2$ for Snow06, $K = 1$ for Yago07, $K = 3$ for TAREC^{MATCH}, and $K = 2$ for both local and inference models of TAREC^{SEARCH}. These values of K are the same on Test-II.[17] This shows that while the best value of K may vary with different systems, it is consistent across the data sets. Hence, we use $K = 2$ for further experiments with TAREC^{SEARCH}, unless specified otherwise.

We do not use special tactics to handle polysemous terms. However, our approaches to building Wikipedia representations for input terms described in Section 3 tie the senses of the two input terms together, thus, implicitly, tend to capture the potential meanings of the terms. We do not use this procedure in Snow06 because WordNet and Wikipedia are different in their structures. We also do not use this procedure in Yago07 because in YAGO, a term is mapped onto the ontology by using the MEANS operator (in Pattern 1, Fig. 4). This cannot follow our procedure.

5.2 Comparison to Harvested Knowledge

As we have discussed earlier, the outputs of bootstrapping-based algorithms is usually limited to a small number of high-quality terms while sacrificing coverage (or vice versa). For example, the full Espresso algorithm (Pantel and Pennacchiotti, 2006) extracted 69,156 instances of *is-a* relation with 36.2% of precision. Similarly,

[17] The best results on Test-II with $K = 2$ and $K = 3$ are similar.

(Kozareva et al., 2008) evaluated only a small number (a few hundreds) of harvested instances. Recently, (Baroni and Lenci, 2010) proposed a general framework for extracting properties of input terms. Their **TypeDM** model harvested 5,000 significant properties for each term out of 20,410 noun mentions. For example, the properties of *marine* include $\langle own, bomb\rangle$, $\langle use, gun\rangle$. Using vector space models we could measure the similarity between terms using their property vectors. However, since the information available in TypeDM does not directly support predicting ancestor relation between terms, we only evaluate TypeDM in classifying sibling vs. no relation. To accommodate this experiment, we develop the following procedure, giving a list of semantic classes.

- For each semantic class, use some seeds to compute a centroid vector from the seeds' vectors in TypeDM.
- Each term in an input pair is classified into its best semantic class based on the cosine similarity between its vector and the centroid vector of the semantic classes.
- Two terms are siblings if they are classified into the same semantic class; and have no relation, otherwise.

Out of the terms in OrgData-I, only 345 terms are covered by the noun mentions in TypeDM. These terms belong to 10 significant semantic classes. For each semantic class, we randomly pick 5 instances as its seeds to compute its single centroid vector. The rest of the overlapping instances are randomly paired to make a data set of 4,000 pairs of terms balanced in the number of sibling and no relation pairs. On this data set, TypeDM achieves an accuracy of 79.75%. TAREC^{SEARCH} (Local), with the local classifier trained on the training set (with 4 taxonomic relation classes) of Dataset-I, gives 78.35% of accuracy. TAREC^{SEARCH} (Inference) system achieves 82.65%. We also re-train and evaluate the local classifier of TAREC^{SEARCH} (Local) on the same training set but without ancestor-relation examples. This local classifier achieves an accuracy of 81.08%.

These results show that although the full system, TAREC^{SEARCH} (Inference), achieves better performance, TypeDM is very competitive in recognizing sibling vs. no relation. It has not been straightforward to apply the TypeDM model to ancestor relations between terms. As a result we only tested it in the limited setting where semantic classes are given in advance.

5.3 Experimental Analysis

In this section, we discuss some experimental analyses to better understand our systems. In all these experiments, TAREC uses the search-based approach to build Wikipedia representation.

Precision and Recall: We study TAREC on individual taxonomic relations using Precision and Recall. Table 7 shows that TAREC (Inference) performs very well on ancestor relations. Sibling and no relation are the most difficult relations to classify. In the same experimental setting on Test-I, Yago07 achieves 79.34% and

Table 7. *Performance of TAREC (Inference) on individual taxonomic relation.*

	Test-I		**Test-II**	
	Prec	Rec	Prec	Rec
$x \leftarrow y$	95.82	88.01	96.46	88.48
$x \rightarrow y$	94.61	89.29	96.15	88.86
$x \leftrightarrow y$	79.23	84.01	83.15	81.87
$x \nleftrightarrow y$	73.94	79.9	75.54	88.27
Average	85.9	85.3	87.83	86.87

Table 8. *Performance of the systems on special data sets, in accuracy. On the non-Wikipedia test set, TAREC (Local) simply returns sibling relation. Note that TAREC uses search-based approach to build Wikipedia representation for input terms.*

System	**Wiki**	**WordNet**	**non-Wiki**
Strube07	24.59	24.13	21.18
Snow06	41.23	46.91	34.46
Yago07	69.95	70.42	34.26
TAREC (Local)	89.37	89.72	31.22
TAREC (Inference)	**91.03**	**91.2**	**45.21**

66.03% of average Precision and Recall, respectively. These numbers on Test-II are 81.33% and 70.44%.

Special Data Sets: We evaluate all systems that use hierarchical structures as background knowledge on three special data sets derived from Test-I. From 12,000 pairs in Test-I, we created a test set, **Wiki**, consisting of $10,456$ pairs with all terms in Wikipedia. We use the rest of $1,544$ pairs with at least one non-Wikipedia term to build a **non-Wiki** test set. The third data set, **WordNet**, contains $8,625$ pairs with all terms in WordNet and Wikipedia. Table 8 shows the performance of the systems on these data sets. Unsurprisingly, Yago07 gets better results on Wiki than on Test-I. Snow06, as expected, gives better performance on the WordNet test set. TAREC (Inference) still significantly outperforms these systems. The improvement of TAREC (Inference) over TAREC (local) on the Wiki and WordNet test sets emphasizes the contribution of the inference model, whereas the improvement on the non-Wikipedia test set shows the contribution of the normalization procedure described in Section 3.4.

Table 9. *TAREC with different sources providing related terms for inference.*

System	K=1	K=2	K=3	K=4
TAREC (Inference)	82.93	85.34	85.23	83.95
TAREC (Gold Inference)	83.46	86.18	85.9	84.93

Contribution of Related Terms in Inference: We evaluate TAREC (Inference) when the inference procedure is fed by related terms that are generated using a "gold standard" source instead of YAGO. To do this, we use the original data which was used to generate Test-I. For each term in the examples of Test-I, we get its ancestors, siblings, and children, if any, from the original data and use them as related terms in the inference model. This system is referred to as **TAREC (Gold Inference)**. Table 9 shows the results of the two systems on different K as the number of levels to go up on the Wikipedia category system. We see that TAREC gets better results when doing inference with better related terms. In this experiment, the two systems use the same number of related terms.

6 Related Work

There are several works that aim at building taxonomies and ontologies which organize concepts and their taxonomic relations into hierarchical structures. (Snow *et al.*, 2005; Snow *et al.*, 2006) constructed classifiers to identify hypernym relationship between mentions from dependency trees of large corpora. Mentions with recognized hypernym relation are extracted and incorporated into a manually constructed lexical database, WordNet (Fellbaum, 1998), resulting in the Extended WordNet, which has been augmented this way with more than $400,000$ synsets. (Ponzetto and Strube, 2007) and (Suchanek *et al.*, 2007) both mined Wikipedia to construct hierarchical structures of concepts and relations. While the former exploited the Wikipedia category system as a conceptual network and extracted a taxonomy consisting of subsumption relations, the latter presented the YAGO ontology, which was automatically constructed by mining and combining Wikipedia structure and information with WordNet. A natural way to use these hierarchical structures to support taxonomic relation classification is to map targeted terms onto the hierarchies and check if they subsume each other or share a common subsumer. However, this approach is limited because constructed hierarchies may suffer from noise and inflexibility in dealing with ambiguous terms.

On the other hand, information extraction bootstrapping algorithms, such as (Pantel and Pennacchiotti, 2006; Kozareva *et al.*, 2008), automatically harvest related terms on large corpora by starting with a few seeds of pre-specified relations (e.g. *is-a*, *part-of*). Bootstrapping algorithms rely on some scoring function to assess the quality of terms and additional patterns extracted during bootstrapping iterations. Similarly, but with a different focus, Open IE, (Banko and Etzioni, 2008;

Davidov and Rappoport, 2008), deals with a large number of relations which are not pre-specified. Either way, the output of these algorithms is usually limited to a small number of high-quality terms while sacrificing coverage (or vice versa). Moreover, an Open IE system cannot control the extracted relations and this is essential when identifying taxonomic relations. Recently, there has been much work on distributional semantic models (DSMs) that leverage the context a word appears in to harvest words based on their semantic similarity in vector spaces (Padó and Lapata, 2007; Turney and Pantel, 2010; Baroni and Lenci, 2010). Especially, (Baroni and Lenci, 2010) described a general framework of DSMs that extracts significant contexts of given terms from large corpora. Consequently, a term can be represented by a vector of contexts in which it frequently appears. Any vector space model could then use the terms' vectors to cluster terms into semantic classes. Sibling terms (e.g. *Honda*, *Toyota*), therefore, have very high chance to be clustered together. Nevertheless, this approach cannot recognize ancestor relations. In this paper, we compare TAREC with this framework only on recognizing sibling vs. no relation, in a strict experimental setting which pre-specifies the semantic classes to which the terms belong.

7 Conclusions

We studied an important component of many computational linguistics tasks: determining taxonomic relations between terms. We have argued that simply looking up the relation of input terms in structured resources cannot support this task well enough, and provided empirical support for this claim. We presented TAREC, a novel algorithmic approach that leverages information from the Wikipedia structure and uses machine learning and a constraint-based inference model to mitigate the noise and the level of uncertainty inherent in these resources. Our experimental study showed that both the local and the global models of TAREC significantly outperform other systems built upon existing well-known knowledge sources. Moreover, our algorithmic approach generalizes and handles well non-Wikipedia terms across semantic classes. Our future work will include an evaluation of TAREC in the context of textual inference applications.

References

Abad, A., Bentivogli, L., Dagan, I., Giampiccolo, D., Mirkin, S., Pianta, E., and Stern, A. 2010. A Resource for Investigating the Impact of Anaphora and Coreference on Inference. *In:* Chair), Nicoletta Calzolari (Conference, Choukri, Khalid, Maegaard, Bente, Mariani, Joseph, Odijk, Jan, Piperidis, Stelios, Rosner, Mike, and Tapias, Daniel (eds), *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA).

Banko, M., and Etzioni, O. 2008. The Tradeoffs Between Open and Traditional Relation Extraction. *Pages 28–36 of: Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics.

Banko, M., Cafarella, M., Soderland, M., Broadhead, M., and Etzioni, O. 2007. Open

Information Extraction from the Web. *Pages 2670–2676 of: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).*

Baroni, M., and Lenci, A. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, **36**, 673–721.

Bishop, C. M. 1996. *Neural networks for pattern recognition.* Oxford, UK: Oxford University Press.

Chakrabarti, S., Dom, B., Agrawal, R., and Raghavan, P. 1997. Using Taxonomy, Discriminants, and Signatures for Navigating in Text Databases. *Pages 446–455 of: Proceedings of the 23rd International Conference on Very Large Data Bases.* VLDB '97. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Chang, M., Ratinov, L., and Roth, D. 2008 (July). Constraints as Prior Knowledge. *Pages 32–39 of: ICML Workshop on Prior Knowledge for Text and Language Processing.*

Dagan, I., Glickman, O., and Magnini, B. (eds). 2006. *The PASCAL Recognising Textual Entailment Challenge.* Vol. 3944. Springer-Verlag, Berlin.

Davidov, D., and Rappoport, A. 2008. Unsupervised Discovery of Generic Relationships Using Pattern Clusters and its Evaluation by Automatically Generated SAT Analogy Questions. *Pages 692–700 of: Proceedings of ACL-08: HLT.* Columbus, Ohio: Association for Computational Linguistics.

Denis, P., and Baldridge, J. 2007. Joint Determination of Anaphoricity and Coreference Resolution using Integer Programming. *Pages 236–243 of: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics.* Rochester, New York: Association for Computational Linguistics.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database.* MIT Press.

Freund, Yoav, and Schapire, Robert E. 1999. Large Margin Classification Using the Perceptron Algorithm. *Machine Learning*, **37**(3), 277–296.

Gabrilovich, Evgeniy, and Markovitch, Shaul. 2007. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. *Pages 1606–1611 of: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).*

Hearst, M. A. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. *Pages 539–545 of: Proceedings the International Conference on Computational Linguistics (COLING).*

Hotho, Andreas, Staab, Steffen, and Gerd, Stum. 2003. Ontologies Improve Text Document Clustering. *Pages 541–544 of: IEEE International Conference on Data Mining(ICDM).* ICDM '03. Washington, DC, USA: IEEE Computer Society.

Kozareva, Z., Riloff, E., and Hovy, E. 2008. Semantic Class Learning from the Web with Hyponym Pattern Linkage Graphs. *Pages 1048–1056 of: Proceedings of ACL-08: HLT.* Columbus, Ohio: Association for Computational Linguistics.

MacCartney, Bill, and Manning, Christopher D. 2008. Modeling Semantic Containment and Exclusion in Natural Language Inference. *Pages 521–528 of: Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1.* COLING '08. Stroudsburg, PA, USA: Association for Computational Linguistics.

Mihalcea, R., and Csomai, A. 2007. Wikify!: linking documents to encyclopedic knowledge. *Pages 233–242 of: Proceedings of ACM Conference on Information and Knowledge Management (CIKM).*

Milne, D., and Witten, I. H. 2008. Learning to link with wikipedia. *Pages 509–518 of: Proceedings of ACM Conference on Information and Knowledge Management (CIKM).*

Paşca, Marius. 2007. Organizing and Searching the World Wide Web of Facts Step Two: Harnessing the Wisdom of the Crowds. *Pages 101–110 of: Proceedings of the 16th international conference on World Wide Web.* WWW '07. New York, NY, USA: ACM.

Paşca, Marius, and Van Durme, Benjamin. 2008. Weakly-Supervised Acquisition of Open-Domain Classes and Class Attributes from Web Documents and Query Logs. *Pages 19–27 of: Proceedings of ACL-08: HLT.* Columbus, Ohio: Association for Computational Linguistics.

Padó, Sebastian, and Lapata, Mirella. 2007. Dependency-Based Construction of Semantic Space Models. *Computational Linguistics*, **33**(June), 161–199.

Pantel, P., and Pennacchiotti, M. 2006. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. *Pages 113–120 of: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).*

Ponzetto, Simone Paolo, and Strube, Michael. 2007. Deriving a large scale taxonomy from Wikipedia. *Pages 1440–1445 of: Proceedings of the 22nd national conference on Artificial intelligence - Volume 2.* AAAI Press.

Punyakanok, V., Roth, D., and Yih, W. 2008. The Importance of Syntactic Parsing and Inference in Semantic Role Labeling. *Computational Linguistics*, **34**(2), 257–287.

Ratinov, L., Downey, D., Anderson, M., and Roth, D. 2011. Local and Global Algorithms for Disambiguation to Wikipedia. *In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).*

Rizzolo, N., and Roth, D. 2010 (May). Learning Based Java for Rapid Development of NLP Systems. *In: Proceedings of the International Conference on Language Resources and Evaluation.*

Roth, D., and Yih, W. 2004. A Linear Programming Formulation for Global Inference in Natural Language Tasks. *Pages 1–8 of:* Ng, Hwee Tou, and Riloff, Ellen (eds), *Proceedings of the Annual Conference on Computational Natural Language Learning (CoNLL).* Association for Computational Linguistics.

Sammons, Mark, Vydiswaran, V. G. Vinod, and Roth, Dan. 2010. "Ask not what textual entailment can do for you...". *Pages 1199–1208 of: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.* Stroudsburg, PA, USA: Association for Computational Linguistics.

Sarmento, L., Jijkuon, V., de Rijke, M., and Oliveira, E. 2007. "More like these": growing entity classes from seeds. *Pages 959–962 of: Proceedings of ACM Conference on Information and Knowledge Management (CIKM).*

Saxena, Ashish Kumar, Sambhu, Ganesh Viswanath, Kaushik, Saroj, and Subramaniam, L. Venkata. 2007. IITD-IBMIRL System for Question Answering Using Pattern Matching, Semantic Type and Semantic Category Recognition. *In: TREC.*

Sekine, S. 2006. On-Demand Information Extraction. *Pages 731–738 of: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).*

Snow, R., Jurafsky, D., and Ng, A.Y. 2005. Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems*, **17**, 1297–1304.

Snow, Rion, Jurafsky, Daniel, and Ng, Andrew Y. 2006. Semantic taxonomy induction from heterogenous evidence. *Pages 801–808 of: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics.* ACL-44. Stroudsburg, PA, USA: Association for Computational Linguistics.

Suchanek, Fabian M., Kasneci, Gjergji, and Weikum, Gerhard. 2007. Yago: a core of semantic knowledge. *Pages 697–706 of: Proceedings of the 16th international conference on World Wide Web.* WWW '07. New York, NY, USA: ACM.

Turney, Peter D., and Pantel, Patrick. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of AI Research*, **37**, 141.

Vikas, O., Meshram, A. K., Meena, G., and Gupta, A. 2008 (June). Multiple Document Summarization Using Principal Component Analysis Incorporating Semantic Vector Space Model. *Pages 141–156 of: Computational Linguistics and Chinese Language Processing*, vol. 13.

Vyas, Vishnu, and Pantel, Patrick. 2009. Semi-automatic entity set refinement. *Pages 290–298 of: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics.* NAACL '09. Stroudsburg, PA, USA: Association for Computational Linguistics.