Abstract

We present a relation extraction system that is specifically designed to extract taxonomic relations from Wikipedia. In contrast with previous related work, the proposed system does not rely on any external knowledge bases – its training examples and feature vector representations are derived exclusively from Wikipedia. Furthermore, the entire system is trained without any manually labeled examples – a diverse set of distantly labeled training examples for IS-A relation classification is extracted automatically from list pages and revision histories. We describe a rich set of feature representations that are derived from salient Wikipedia structures, as well as from syntactic and semantic processings of text. A bootstrapped version of the system is used to create additional features for itself, leading to improvements in extraction accuracy. The taxonomic relation extraction system is applied on the Wikipedia category graph and the results are stored in a graph database, with associated confidence levels that allow trading off recall vs. precision. Because it depends exclusively on Wikipedia and requires no manual labeling, the extraction system will be straightforward to update with newer versions of Wikipedia.

1 Distant Wikipedia Supervision

The taxonomic relation extraction system is trained as a binary classifier on a dataset of training examples that are extracted automatically from Wikipedia itself. Positive and negative examples are created by exploiting two very different Wikipedia structures: list pages and revision history.

1.1 Distant Positive Examples

We used list pages in Wikipedia as a source of positive examples of IS-A pairs. A list page has a title List of \(<X>\) and contains a set of links \(<Y>\) organized in a list or in a table. For example, the page List of cocktails contains a multi-level list of links to articles such as The Modernista or Black and Tan, whereas the page List of antibiotics contains a multi-heading table in which the first column contains links to titles such as Amoxicilin and Penicilin G. Given a List of \(<X>\) title and links \(<Y>\) contained in the list page, we create positive example pairs \((P, C)\) as follows:
1. As possible parent arguments $P$, we used all the base noun phrases that were textually contained in the noun phrase $X$ and that have the same head as $X$. For example, if the list page has the title List of dog diseases, we consider as candidate categories both dog diseases and diseases. Of these candidate categories, we use as parent categories $P$ only those that satisfy the following constraints:

(a) The category name corresponds to an actual category page in Wikipedia e.g., Category::Diseases or Category::Dog Diseases.

(b) The corresponding category page has a main article associated with it. For the example above, only the candidate category diseases survives this step, since its associated category page in Wikipedia contains the template text the main article for this category is disease.

2. As possible child arguments $C$, we use the links $Y$ extracted from the list page content as follows:

- If the list page contains an itemized list, we extract all the links that appeared at the beginning of list items. Thus, we do not extract the list item Absinthe as a candidate child argument since it is not a link, but we extract the link The Modernista from one level below.
- If the list page contains a table, we extract all the links appearing in the first column in the table, as long as the column heading is a noun phrase with the head name, or a noun phrases that has the same head as the list topic $X$. For example, for the List of antibiotics page, the heading of the first column in the table is Generic names, therefore we extract all the links in that column as potential child arguments.

There are 52,332 list pages in Wikipedia, containing a total of 4,253,551 links. Given the large number of candidate positive pairs, we seek to further increase the quality of the extracted pairs $\langle P, C \rangle$ by running them through a set of filters, as follows:

1. The pair is kept as a positive example only if $C$ is also a direct descendant of $P$ in the Wikipedia category graph. This is based on the observation that listing a title $C$ on $P$’s list page and connecting the same title with $P$’s category page provide a strong joint indication that $C$ is in an IS-A relationship with $P$.

2. The pair is eliminated if $C$ is a redirect page, as redirect relationships are open ended.

3. Ignore list pages with fewer than 5 extracted pairs.

4. Ignore pairs where either the parent or the child arguments link to stub pages.

Upon applying the filters above, we obtained 2,970 parent categories $P$, for a total of 144,791 positive pairs $\langle P, C \rangle$. In order to keep training times feasible while maintaining diversity, we sample the same number of positive pairs for each of the extracted parent categories, as described in more detail in Section 4.

1.2 Distant Negative Examples

Negative taxonomic examples are not as easy to find as positive examples in Wikipedia. First, there is no equivalent of list pages for negative examples that would directly specify that a page $Y$ cannot be an element in a list of items of type $X$. Furthermore, since Wikipedia lists are not comprehensive, the fact that a page $Y$ is not linked from a list page List of $X$ does not necessarily imply that the pair $\langle X, Y \rangle$ is a negative IS-A example. However, for a randomly chosen $Y$ in Wikipedia that is not linked from the list page List of $X$, the likelihood that $Y$ is subsumed by $X$ is generally very small. Therefore, a training dataset containing such distant negative examples would have a relatively low number of false negatives, which would not have much impact on a machine learning algorithm that is tolerant to noise, assuming a sufficiently large number of training examples are available. There is however one essential problem with this approach: since randomly chosen pages are unlikely to belong to a semantic relationship, the trained classifier would likely have poor performance on negative test pairs $\langle P, C \rangle$ where title $C$ has a relationship, other than IS-A, with title category $P$. This would be the case, for example, if the trained classifier were used to extract a taxonomic subgraph from the Wikipedia category graph, as a pair $\langle P, C \rangle$ is included in the category graph only based on an existing relationship between the argument titles in the pair.
It is therefore important that the set of negative examples contains pairs of titles that belong to a salient semantic relation, of the type that leads to inclusion in the Wikipedia category graph. Take, for example, the title $C = \text{Principia Mathematica}$ which is included in the category $P = \text{Mathematical logic}$ in Wikipedia. While there is no IS-A relationship between them, the two titles are not entirely unrelated either. In order to obtain pairs of related titles that are not in an IS-A relationship, we used the revision history in Wikipedia as follows:

1. Collect pairs $\langle P, C \rangle$ such that $C$ is listed under category $P$ in some version of Wikipedia before 2012-07-03, but $C$ is no longer under category $P$ in the reference 2012-07-03 snapshot of Wikipedia. Because at one time the titles $\langle P, C \rangle$ were linked in the category graph, we infer that it is very likely that there is some relationship between them. Furthermore, since they are no longer linked in the category graph, it is unlikely they are in an IS-A relationship. For example, the title $C = \text{Sterndrive}$ used to be under the category $P = \text{Boats}$, which indicates they are related semantically, but the pair was eventually removed because it no longer appears in the category graph in the reference version. This step results in 312,896 candidate negative pairs.

2. Some category links are removed in one Wikipedia version, only to be added back later e.g., in cases of vandalism or simply mistakes. To control for these transient changes, from the set of pairs extracted at step 1 above we remove those pairs that were put back in the category graph anytime between the reference 2012-07-03 snapshot of Wikipedia and the later 2014-02-03 snapshot.

3. As in the case of positive examples, whenever one of the arguments in the candidate pair $\langle P, C \rangle$ is a category page, we keep the pair only if the category page has a main title page associated with it. Furthermore, we filter out pairs that contain stub pages.

4. Pages $C$ may be removed from a parent category $P$ in order to be linked to a more specific subcategory of $P$. Such pairs are filtered out automatically from the set of distant negative examples by verifying that the two argument titles in $\langle P, C \rangle$ are not in a descendant relationship in the Wikipedia category graph. This verification step is performed on both the reference 2012 snapshot and the future 2014 snapshot of Wikipedia, by considering paths starting at $C$ and extending up to 10 edges towards the root of the category graph.

The last 3 steps reduced the negative portion of the dataset to 19,209 negative pairs.

## 2 Manually Labeled Dataset

The set of articles and categories in Wikipedia form a directed acyclic graph in which inner nodes are categories and outer nodes are articles. We selected the following 10 diverse categories from near the top of the hierarchy as root nodes for a dataset generator algorithm: Artists, Awards, Beverages, Corporations, Films, Natural disasters, Restaurants, Sports clubs, Stars, and Viruses. For each of the 10 root categories, we generated a subgraph dataset, using the algorithm shown in Table 1. Since manually annotating all the descendant categories and articles under the 10 root categories is unfeasible, the purpose of this algorithm is to generate a subgraph for each root category that preserves as much as possible the distribution of the original descendant categories and articles. The algorithm generates random paths in the Wikipedia category graph that start at the root category node $R$ and end with a random descendant article. Each path is initialized with the root category, and at each step its length grows by appending to it a random node selected from the current node’s children $N.C$. Then the random node becomes the current node and the path terminates whenever the current node has no children (i.e., it is an article node). The overall path generation process stops when the set $P.A$ of article nodes contained in the generated paths reaches a predefined size $S$. Due to the fact that an article or category node may have multiple parent categories, two or more different paths may terminate at the same article, which means that the number of randomly generated paths may be greater than the number of articles ending the paths. The random paths thus generated correspond to a subgraph of the original Wikipedia category graph rooted at category $R$. The actual implementation of the algorithm in Table 1 is further optimized to avoid considering nodes whose descendants have all been selected in the previous paths, and consequently to stop when all descendant articles have been selected (to cover the unlikely cases when a root category has fewer than $S$ descendant articles). Articles and
categories that are marked as problematic (e.g. cleanup) or incomplete (e.g. stub) are automatically filtered out from the generated dataset.

Table 1: Dataset generator algorithm.

Using the algorithm in Table 1 we generated 10 subgraphs, one for each of the 10 root categories. Table 2 below shows the overall distribution of nodes with respect to their depth in their respective subgraph. Since each subgraph is a directed acyclic graph, for any given node its depth was considered to be the length of the shortest path between that node and the root category.

With the exception of the root category node, each node in the 10 subgraphs was annotated for IS-A relations with respect to its parent nodes. One important principle that was followed during annotation was that the information in Wikipedia takes precedence over information from any other sources and therefore the annotation of subsumption relations should be consistent with Wikipedia. A supernova, for example, is defined in Wikipedia as “a stellar explosion that is more energetic than a nova”, whereas in WordNet its gloss is “a star that explodes and becomes extremely luminous in the process”. Because a stellar explosion is not really a type of star, giving precedence to Wikipedia meant that Supernova was not labeled as IS-A with respect to the root category Stars. The meaning of a category title was determined based on the main article associated with the category, whereas the meaning of an article title was determined based on the text of the article. After eliminating categories without main articles and list pages, we obtained 508 samples pairs ⟨P,C⟩ out of which 305 were annotated as having an IS-A relationship.

3 Feature Engineering

Determining whether a pair of entities ⟨P,C⟩ are in an IS-A relationship is expressed a a binary classification problem. The pair is represented as a vector of features φ(P,C) and given as input to a binary classification model. We assume that both entities in the pair can be mapped to a corresponding main article in Wikipedia. If the task is to extract a taxonomy from the Wikipedia category graph, P will always be a category page, whereas C can be either a category page or a main article. Most categories in Wikipedia contain the cat main template that indicates a main article associated with the category – for example, the category page Category:Science contains the wiki template {{cat main|Science}}, which is rendered in HTML as The main article for this category is Science, indicating that Science is the title of main article that describes the meaning of the category with the same name. We consider category pages that do not have a corresponding main article in Wikipedia as categories that lack a clear definition. In general, such categories, since they lack a clear definition, have a meaning that cannot be circumscribed exactly and therefore it can be difficult even for humans to determine whether they stand in an IS-A relationship with other
categories or main articles in Wikipedia. Therefore, we decided to train and test the binary classification algorithm only on pairs of entities that are either main articles or that can be mapped to main articles in Wikipedia.

Many of the features described in this section depend on a measure of the semantic similarity between words. Given two arbitrary words \( w_1 \) and \( w_2 \), we compute their semantic similarity \( \text{sim}(w_1, w_2) \) using one of the 3 methods below:

1. **Lexical Similarity**: this corresponds to setting \( \text{sim}(w_1, w_2) = 1[w_1 = w_2] \) i.e. the similarity is 1 if the words are the same, otherwise 0.
2. **Distributional Similarity**: this corresponds to setting \( \text{sim}(w_1, w_2) = \text{we}(w_1)^T \cdot \text{we}(w_2) \), where \( \text{we}(w) \) is a word embedding associated with the word \( w \). We use the Stanford word embeddings, computed using the method described in [1].
3. **Contextual Distributional Similarity**: this corresponds to setting \( \text{sim}(w_1, w_2) = \text{we}(w_1, c_1)^T \cdot \text{we}(w_2, c_2) \), where the word embedding depends both on the word \( w_i \) and its context \( c_i \). The context is defined as a window of 10 words around the word, and is used according to the method described in [1].

Since the aim of this project is to extract a taxonomy that reflects the knowledge encoded in Wikipedia exclusively, the features in \( \phi(P, C) \) will be computed only from information contained in Wikipedia. The major types of Wikipedia structures used to derive features are as follows:

1. **Links**: links to articles, categories, templates, or photos.
2. **Format**: types of fonts (regular, bold, or italic), section headings, list items or column headings and cell contents in tables.
3. **Templates**: the templates that are used in a main article to specify higher level, structured information about the topic.
4. **Titles**: the titles associated with the corresponding main articles in Wikipedia.
5. **Infoboxes**: the fixed-format tables placed in the top right corner of an article, designed to present a summary of definitive aspects of the article. These aspects are normally general enough to be reused by other related articles.
6. **Text**: the textual content of the article.

### 3.1 Link Features

The hyperlinks in Wikipedia interconnect main articles, categories, photos, templates, or infoboxes in a complex graph in which the set of nodes is dominated by main articles and categories. The main types of links in Wikipedia are those that connect articles to other articles. The presence of a directed link between two pages \( P \) and \( C \) may indicate a relationship between the two entities. Furthermore, the number of shared links between \( P \) and \( C \) can be used as a proxy for the strength of their relationship. Let \( A.links \) be the set of hyperlinks to other articles contained by an article \( A \). Correspondingly, we create the following hyperlink features:

- \( \phi_1(P, C) = 1 \) if article \( P \) contains links to \( C \), i.e., \( C \in P\text{.links} \).
- \( \phi_2(P, C) = 1 \) if article \( C \) contains links to \( P \), i.e., \( P \in C\text{.links} \).
- \( \phi_3(P, C) = J(P\text{.links}, C\text{.links}) \), where \( J(S, T) = |S \cap T| / |S \cup T| \) is the Jaccard similarity between the two sets.

Pages that are in a taxonomic relationship often have photos in common. For example, the articles \( \text{Sun} \) and \( \text{Star} \) both link to the same photo of the Sun. Let \( A\text{.photos} \) be the set of photo links contained by a main article \( A \). Correspondingly, we define the following photo similarity feature:

- \( \phi_4(P, C) = 1[P\text{.photos} \cap C\text{.photos} \neq \emptyset] \), i.e., the feature is 1 if the articles share at least one photo.

Similarly, pages that are in a taxonomic relationship often have templates in common. Let \( A\text{.temps} \) be the set of templates used by article \( A \). Correspondingly, we define the following template similarity feature:
• \( \phi_0(P,C) = J(P.temps, C.temps) \).

A main article is usually associated with one or more categories in Wikipedia through category links. Similarly, categories may be associated with their own higher level categories. Articles that are in a taxonomic relationship are expected to share many of their parent or ancestor categories in the Wikipedia category graph. Let \( A.cats.i \) denote the set of ancestor categories that are up to \( i \) levels above article \( A \) in the category graph. Thus, \( A.cats.1 \) denotes the set of direct parent categories of \( A \), whereas \( A.cats.2 \) denotes the parent and grandparent categories of \( A \). Correspondingly, we define the following category similarity features:

• \( \phi_0(P,C) = J(P.cats.1, C.cats.1) \).
• \( \phi_7(P,C) = J(P.cats.2, C.cats.2) \).
• \( \phi_8(P,C) = J(P.cats.3, C.cats.3) \).

### 3.2 Format Features

Articles that are in a taxonomic relationship often make one or more textual mentions to the related entity. These mentions are often emphasized, either through font styles such as bold or italic, or through formatting structures such as section headings, lists, and tables.

Let \( A.title \) denote the title of article \( A \); let \( A.fonts \) denote the phrases in \( A \) that have been emphasized in bold or italic; let \( A.heads \) denote the section headings of \( A \); let \( A.litems \) denote the list items in \( A \); and let \( A.tcells \) denote the table entries in \( A \). Correspondingly, we define features that check whether one entity in the pair is emphasized in the other entity article, as follows:

• \( \phi_9(P,C) = 1 [P.title \in C.fonts] \), i.e., the feature is 1 if the title of \( P \) was emphasized using bold or italic fonts somewhere in the article \( C \). We also define the symmetric feature \( \phi_{10}(P,C) = 1 [C.title \in P.fonts] \).
• \( \phi_{11}(P,C) = 1 [P.title \in C.heads] \), i.e., the feature is 1 if the title of \( P \) appeared as a section heading in the article \( C \). We also define the symmetric feature \( \phi_{12}(P,C) = 1 [C.title \in P.heads] \).
• \( \phi_{13}(P,C) = 1 [P.title \in C.litems] \), i.e., the feature is 1 if the title of \( P \) appeared as a list item in the article \( C \). We also define the symmetric feature \( \phi_{14}(P,C) = 1 [C.title \in P.litems] \).
• \( \phi_{15}(P,C) = 1 [P.title \in C.tcells] \), i.e., the feature is 1 if the title of \( P \) appeared as a cell in a table in the article \( C \). We also define the symmetric feature \( \phi_{16}(P,C) = 1 [C.title \in P.tcells] \).

Similarly with some of the link features above, we expect articles that are in an IS-A relationship to refer to a set of common entities. Correspondingly, we define the following co-citation features:

• \( \phi_{17}(P,C) = J(P.fonts, C.fonts) \).
• \( \phi_{18}(P,C) = J(P.heads, C.heads) \).
• \( \phi_{19}(P,C) = J(P.litems, C.litems) \).
• \( \phi_{20}(P,C) = J(P.tcells, C.tcells) \).

### 3.3 Template Features

A strong signal for a potential taxonomic relationship is provided by the about template. When included in the wiki source of an article, the about template generates a disambiguating sentence right before the main textual content. The sentence describes what the article is about, and is generated especially in cases where the article title is polysemous. For example, the article Beer contains the template \{\{About|the alcoholic beverage\}\}, which is rendered in HTML as This article is about the alcoholic beverage. This gives a strong indication that there is an IS-A relationship between Beer and Alcoholic beverages. Correspondingly, we create the following template-based similarity features:
If an article $C$ contains a template whose name matches the singular or plural version of $P$’s title.

### 3.4 Title Features

Each article in Wikipedia is uniquely referenced by a title, consisting of one or more words separated by spaces or underscores and occasionally a parenthetical explanation. For example, the article for the entity Turing that refers to the English computer scientist Alan Turing, whereas the article on Turing with the stream cipher meaning has the unique title Turing (cipher). Let $A.paren$ denote the parenthetical phrase associated with the title of article $A$. Since the parenthetical phrase usually indicates a strong taxonomic relationship, we create the following parenthetical similarity feature:

$\phi_{22}(P, C) = 1$ if the article $C$ contains a template whose name matches the singular or plural version of $P$’s title.

The title of a Wikipedia article can be seen as a very compressed representation of its content. As such, article titles offer one of most salient signals for a potential taxonomic relationship between entities in Wikipedia. For example, if the titles of two articles share the same syntactic head, it is very likely that the two articles are in a close semantic relationship. Furthermore, the likelihood that this is a taxonomic relationship is increased if one of the titles is included in the other title, as in the pair $(P, C) = \text{Northern pole stars}, \text{Pole stars}$. Letting $A.head$ denote the syntactic head of $A.title$, we define the following title based lexical similarity features:

$\phi_{24}(P, C) = 1[\text{P.title} = \text{C.paren}]$, i.e., $C$’s parenthetical phrase is the same as $P$’s title, as in the pair Rocky (film series) and Film series.

When the lexical heads do not match and $P$ is a category page, an alternative way of determining semantic subsumption is to look at the syntactic heads of the pages under category $P$. Let $A.tophead$ denote the most frequent head of all the page titles under category $A$ in the Wikipedia category graph. Correspondingly, we define another head matching feature as follows:

$\phi_{28}(P, C) = 1[\text{C.head} = \text{P.tophead}]$, i.e., $C$’s syntactic head is the most frequent head of page titles under category $P$.

If the two titles do not share the same syntactic head, the semantic similarity between them could be another useful indicator for the IS-A relationship. Correspondingly, we define the following title-based distributional similarity features:

$\phi_{29}(P, C) = \text{dsim}(P.head, C.head)$, i.e., the distributional similarity between the two syntactic heads.

$\phi_{30}(P, C) = \text{simTitle}(P.title, C.title)$, i.e., the distributional similarity of the two titles, computed with the greedy procedure shown in Algorithm[1] below. The distributional similarity between the two syntactic heads is multiplied with the distributional similarity of the two most similar words in the two titles, other than the two heads. The most similar words are then removed from the titles, and the procedure continues until one of the two titles
becomes empty. By multiplying word similarities into the overall title similarity measure we ensure that we obtain a low similarity score whenever the two titles contain words that are too dissimilar.

Algorithm 1 \texttt{simTitle}(T_1, T_2)

\textbf{Input:} Two Wikipedia titles $T_1$ and $T_2$. A distributional similarity measure $dsim$.

\textbf{Output:} A similarity score $\text{simTitle}(T_1, T_2)$ between the two titles.

1: $\text{simTitle}(T_1, T_2) \leftarrow \text{dsim}(T_1.\text{head}, T_2.\text{head})$

2: $T_1.\text{rest} \leftarrow T_1 - \{T_1.\text{head}\}$

3: $T_2.\text{rest} \leftarrow T_2 - \{T_2.\text{head}\}$

4: \textbf{while} $|T_1.\text{rest}| > 0$ \textbf{and} $|T_2.\text{rest}| > 0$ \textbf{do}

5: \hspace{1em} $t_1, t_2 = \text{arg max}_{t_i \in T_i.\text{rest}} \text{dsim}(t_1, t_2)$

6: \hspace{1em} $\text{simTitle}(T_1, T_2) \leftarrow \text{simTitle}(T_1, T_2) \ast \text{dsim}(w_1, w_2)$

7: \hspace{1em} $T_1.\text{rest} \leftarrow T_1.\text{rest} - \{t_1\}$

8: \hspace{1em} $T_2.\text{rest} \leftarrow T_2.\text{rest} - \{t_2\}$

9: \hspace{1em} \textbf{return} $\text{simTitle}(T_1, T_2)$

3.5 Infobox Features

According to Wikipedia\footnote{http://en.wikipedia.org/wiki/Help:Infobox}, “an infobox is a fixed-format table designed to be added to the top right-hand corner of articles to consistently present a summary of some unifying aspect that the articles share and sometimes to improve navigation to other interrelated articles”. Infoboxes are instances of infobox templates which “are like fact sheets, or sidebars, in magazine articles. They quickly summarize important points in an easy-to-read format”. An infobox template is simply a collection of infobox parameters, whereas an actual infobox is an instance of the infobox template that specifies values for the parameters. Table\[3\] shows the infobox template \texttt{Video Game}, as instantiated in the article \texttt{Global Agenda}.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer(s)</td>
<td>Hi-Rez Studios</td>
<td>Genre</td>
<td>Sitcom</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Publisher(s)</td>
<td>Hi-Rez Studios</td>
<td>Created by</td>
<td>Chuck Lorre</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine</td>
<td>Unreal Engine 3</td>
<td>Directed by</td>
<td>Bill Prady</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Platform(s)</td>
<td>Microsoft Windows</td>
<td>Theme music composer</td>
<td>Mark Cendrowski</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release date</td>
<td>February 1, 2010</td>
<td>Country of origin</td>
<td>Barenaked Ladies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Genre(s)</td>
<td>Massively multiplayer online game</td>
<td>Starring</td>
<td>United States</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mode(s)</td>
<td>Third-person shooter</td>
<td></td>
<td>Johnny Galecki</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Action RPG</td>
<td></td>
<td>Jim Parsons</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Multiplayer</td>
<td></td>
<td>Kaley Cuoco</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Infoboxes instantiated in the articles \texttt{Global Agenda} and \texttt{The Big Bang Theory}.

Let $A.ibox$ be the set of $\langle \text{parameter}, \text{value} \rangle$ pairs instantiated in the infobox shown in the article (empty if the article does not contain an infobox). If two taxonomically related articles contain infoboxes, we expect that they share parameter values. Correspondingly, we define the following infobox similarity feature:

- $\phi_{31}(P, C) = J(P.ibox, C.ibox)$.

An infobox parameter can be seen as a relation linking the article containing the infobox with the parameter value. Some infobox parameters, such as \textit{genre}, \textit{type}, or \textit{family}, indicate a strong taxonomic relationship. For example, the parameter \textit{genre} underlined in Table\[3\] above links pairs of articles that are in an IS-A relationship, e.g., \texttt{Global Agenda} with \texttt{Third-person shooter}, and \texttt{The
Big Bang Theory with the category Sitcom. In order to determine which infobox parameters are more likely to indicate a taxonomic relationship, we use the set of distant positive examples $\mathcal{P}$ that was described in Section 1.1, as follows:

1. For each infobox parameter $ip$, let $\text{count}(ip, \mathcal{P})$ be the number of distant positive pairs $(P, C$) for which $P$ and $C$ are linked by that parameter. Also, let $\text{count}(ip)$ be the total number of pairs of Wikipedia pages $(P, C)$ that are linked by $ip$.

2. We call *taxonomic parameters* the set $\mathcal{T_P}$ of infobox parameters $ip$ that have a non-zero support in the the set of distant positive examples, i.e., parameters $ip$ for which $\text{count}(ip, \mathcal{P}) > 0$.

3. For each taxonomic parameter $ip$, we compute a scaled score $f(ip) \in [0, 1]$ using one of the two normalization strategies below:

\[
\begin{align*}
\text{(a)} & \quad f(ip) = \frac{\text{count}(ip, \mathcal{P})}{\max_{ip' \in \mathcal{T_P}} \text{count}(ip', \mathcal{P})}, \\
\text{(b)} & \quad f(ip) = \frac{\text{count}(ip, \mathcal{P})}{\text{count}(ip)}.
\end{align*}
\]

Let $A.tps$ denote all taxonomic parameters that appear in an infobox inside an article $A$; let $tp.value(A)$ denote the parameter value associated with a parameter $tp \in A.tps$; and let $\mathcal{T_P}(C, P)$ be a predicate that is true iff the parameter $tp$ links article $C$ with article $P$. Correspondingly, given an input pair of articles $(P, C)$, we define the following taxonomic parameter features:

- $\phi_{32}(P, C, tp) = f(tp)$, for all $tp \in C.tps$ such that $\mathcal{T_P}(C, P)$ is true, i.e., for each taxonomic parameter that links $C$ with $P$, define a feature whose value is the taxonomic score of that parameter.

- $\phi_{33}(P, C) = \sum_{tp \in C.tps} \phi_{32}(P, C, tp)$, i.e., sum the scores of all taxonomic parameters that link $C$ with $P$.

- We also define the symmetric features:
  - $\phi_{34}(P, C) = \phi_{32}(C, P)$.
  - $\phi_{35}(P, C) = \phi_{33}(C, P)$.

### 3.6 Text Features

The topic of a Wikipedia article is often mentioned in the text of the article using coreferential noun phrases. One common way to refer to the topic $C$ inside its article is through its category. Therefore, if the title of a main article $P$ matches one of the noun phrases that are coreferential with $C$, this is a strong indicator that the pair $(P, C)$ is in a taxonomic relationship. Since solving coreference between noun phrases is difficult in general, we only consider two types of patterns that lead to precise coreference extractions: *predicate nominals* and *appositive constructions*. Predicate nominals are common in the first sentence of a Wikipedia article, being used to succinctly define the topic, as in the example below:

**Loyalsock Creek** is a 64-mile-long (103 km) tributary of the West Branch Susquehanna River located chiefly in Sullivan and Lycoming counties in Pennsylvania in the United States.

In this sentence, Tributary is a predicate nominal of Loyalsock Creek, which can be used as a strong signal to infer an IS-A relation between them. Furthermore, whenever the predicate nominal is hyperlinked to its own article, we chain extract the predicate nominal relation from that article too to infer another IS-A relationship, based on transitivity. Consider the first sentence from the Tributary article below:

A tributary or affluent is a stream or river that flows into a main stem (or parent) river or a lake.

In this sentence, River is a predicate nominal of Tributary. Since Tributary itself was a predicate nominal of Loyalsock Creek, we can link River to Loyalsock Creek through a chain of two predicate nominal relations. We use chains of up to 3 predicate nominal relations to create the following coreference features:
• \( \phi_{36}(P, C) = 1 \) if \( P \) is a predicate nominal of \( C \) through a chain of length 1, i.e., a direct predicate nominal.
• \( \phi_{37}(P, C) = 1 \) if \( P \) is a predicate nominal of \( C \) through a chain of length 2.
• \( \phi_{38}(P, C) = 1 \) if \( P \) is a predicate nominal of \( C \) through a chain of length 3.
• \( \phi_{39}(P, C) = 1 \) if the predicate nominals if the first sentences of articles \( P \) and \( C \) are the same.

We create additional features that go beyond the simple lexical matching used in the feature above. Let \( A.pn \) denote the predicate nominal in the first sentence of article \( A \). Similar with feature \( \phi_{28} \) introduced earlier, we create a feature that matches the predicate nominal of \( C \) with the most frequent head of page titles listed under category \( P \).

• \( \phi_{40}(P, C) = 1 \) if \( P.tophead = C.pn.head \) if \( P.tophead \) matches the head of the predicate nominal of the first sentence in \( C \).

When simple lexical matching fails, we try to match predicate nominals using their contextual distributional similarity:

• \( \phi_{41}(P, C) = dsim(P.pn, C.pn) \), where \( dsim \) measure distributional similarity in the first sentence context.

Furthermore, we create additional features that exploit coreference links beyond the first sentence in an article, as follows:

• \( \phi_{42}(P, C) = 1 \) if the second sentence of \( C \) begins with a definite noun phrase as a subject, and the syntactic head of the subject matches the head of the title of \( P \).
• \( \phi_{43}(P, C) = 1 \) if the predicate nominal of \( C \) is coreferent with a noun phrase whose syntactic head matches the head of \( P \)’s title.
• \( \phi_{44}(P, C) = 1 \) if the titles of \( P \) and \( C \) are in an appositive pattern anywhere in \( C \)’s article.
• \( \phi_{45}(P, C) = 1 \) if the titles of \( P \) and \( C \) are linked in the syntactic dependency graph of a sentence inside \( C \)’s article through a preposition other than “as”. This feature indicates non-coreference between the two titles and is therefore a good indication that the two titles are not in an IS-A relationship.

As an example of feature \( \phi_{46} \), consider the pair of articles Malta Jazz Festival and Jazz festivals. The first article starts with the two sentences below:

*The Malta Jazz Festival* is a three day musical event staged held every July on the Mediterranean island of Malta, organised by the Malta Council for Culture and the Arts. The festival has been held annually since 1990 at Ta’ Liesse on the Valletta waterfront.

The noun festival is coreferent with *The Malta Jazz Festival* and it also matches the head of Jazz festivals.

The sentences in the first paragraph of a Wikipedia article function as a concise definition of the main entity, or as an abstract of the topic covered in the article. We tokenize the first paragraph, remove stopwords, and reduce the content words to their lemmas. Let \( A.wsent.n \) and \( A.nsent.n \) be the bag of words and nouns, respectively, from the first \( n \) sentences in the article \( A \). Correspondingly, we define features that capture the lexical and semantic similarity of the text contained in the first paragraphs, as follows:

• \( \phi_{47}(P, C) = tsim(P.wsent.1, C.wsent.1) \), i.e., the bag-of-words similarity between the first two sentences.
• \( \phi_{48}(P, C) = vsm(P.wsent.1, C.wsent.1) \), i.e., the cosine similarity of the tf.idf word vectors of the first sentences from each article.
• \( \phi_{49}(P, C) = vsm(P.wsent.3, C.wsent.3) \), i.e., the cosine similarity of the tf.idf word vectors of the first three sentences from each article.
• \( \phi_{50}(P, C) = vsm(P.nsent.1, C.nsent.1) \), i.e., the cosine similarity of the tf.idf noun vectors of the first sentences from each article.
• $\phi_{q1}(P, C) = vsm(P.nsent.3, C.nsent.3)$, i.e., the cosine similarity of the tf.idf noun vectors of the first three sentences from each article.

To compute feature $\phi_{q7}$ above, we use the greedy text-to-text similarity function proposed by Mihalcea et al. [2] as follows:

$$t\text{sim}(W_1, W_2) = \frac{\sum_{w_1 \in W_1} maxSim(w_1, W_2) * idf(w_1) + \sum_{w_2 \in W_2} maxSim(w_2, W_1) * idf(w_2)}{\sum_{w_1 \in W_1} idf(w_1) + \sum_{w_2 \in W_2} idf(w_2)}$$

(1)

For each word $w_1 \in W_1$, $maxSim(w_1, W_2)$ computes the maximum distributional semantic similarity between $w$ and any word $w_2 \in W_2$:

$$maxSim(w_1, W_2) = \arg\max_{w_2 \in W_2} dsim(w_1, w_2)$$

(2)

### 3.6.1 Dependency Patterns

The precise coreference patterns – *predicate nominals* and *appositive constructions* – used in this section can be seen as only two of a larger set of syntactic dependency patterns that are indicative of the IS-A relationship. Consider, for example, the first sentence of the article *Helix*:

*A helix is a type of smooth space curve.*

The predicate nominal in this sentence is *type*, however the actual IS-A relationship is between $C = \text{Helix}$ and $P = \text{Space curve}$. The dependency pattern between the two titles in the first sentence is $C \rightarrow \{i\} \rightarrow \text{type} \leftarrow \text{of} \leftarrow P$. Let $D_P$ be a dependency patterns database that contains this and other patterns indicative of taxonomic relationships; let $f(dp)$ be a score that captures how indicate a pattern $dp$ is for an IS-A relationship; and let $dp(C, P)$ denote the fact that $C$ and $P$ are connected through a dependency pattern $dp$ in Wikipedia. Correspondingly, we define the following generalized pattern features:

• $\phi_{q2}(P, C, dp) = f(dp)$, for all $dp \in D_P$ such that $dp(C, P)$ is true, i.e., define a separate feature for each dependency pattern indicative of a taxonomic relationship that links the two titles somewhere in their Wikipedia articles and set it to the score of that pattern.

• $\phi_{q3}(P, C) = \sum_{dp \in D_P} \phi_{q2}(P, C, dp)$, i.e., sum the scores of all dependency patterns in $D_P$ that link the two articles.

We leverage the set of distant positive examples $P$ described in Section 1.1 to automatically mine the dependency patterns database $D_P$, as follows:

1. For each distant positive pair $\langle P, C \rangle \in P$, find all dependency patterns that link the titles of $C$ and $P$ in the same sentence, anywhere in their articles.
2. Add the dependency patterns found above to the dependency patterns database $D_P$. Each dependency path pattern is a unique combination of lemmas and part-of-speech tags of the words in the dependency path as well as the dependency types of all edges along the dependency path.
3. For each dependency pattern $dp \in D_P$, count the number of times $f(dp)$ that the pattern was found to link distant positive pairs $\langle P, C \rangle \in P$ in the corresponding Wikipedia articles.
4. Scale all pattern scores such that $f(dp) \in [0, 1], \forall dp \in D_P$.

The mining approach above resulted in around 7,000 distinct dependency patterns. To create the dependency pattern features, we used only the 968 patterns that occurred at least 100 times in the distant positive dataset, i.e., patterns that had an unnormalized score above 100.

### 3.7 Bootstrap Features

If an article $C$ is linked to a category $P$ in the Wikipedia category graph, this generally indicates that $C$ is relevant for the topic of the category $P$. Many categories in Wikipedia correspond to
entities that cannot really have other subcategories or entities in an IS-A relationship with them. For example, the category page Category:United States list subcategories and titles such as Category:American society and History of the United States that do not have an IS-A relation with the entity United States. Consequently, when appearing in the $P$ position in a pair $⟨P, C⟩$, such pages should lead to a negative IS-A classification, irrespective of the second argument in the pair. We call the main articles corresponding to such pages instance articles.

Since a classification of articles into instance and non-instance categories is not available in Wikipedia, we use the taxonomical relation classifier itself to perform this classification automatically. Thus, let $Φ(P, C) = [φ_1, φ_2, ..., φ_{53}](P, C)$ be the vector of all features introduced so far. Using this feature representation, we train an SVM classifier on the distant Wikipedia supervision, as described in Section 4. Given an article $P$ that has a category with the same name, let $P.subs$ be the set of articles $C$ that are linked to category $P$ in the Wikipedia category graph. Furthermore, let $svm(Φ(P, C))$ be the margin score that the trained SVM model assigns to an input pair $⟨P, C⟩$. Correspondingly, we create the following instance score features:

- $φ_54(P, C) = 1$ if $\left( \sum_{C' \in P.subs} 1[svm(Φ(P, C')) < 0] \right) > \frac{|P.subs|}{2}$, i.e., the feature is set to 1 if more than half of the pages under category $P$ are classified as not being in an IS-A relationship with the main article $P$. This feature corresponds to a hard instance classification of $P$.
- $φ_{55}(P, C) = \frac{\sum_{C' \in P.subs} 1[svm(Φ(P, C')) < 0]}{|P.subs|}$, i.e., the feature is set to the percentage of pages under category $P$ that are classified as not being in an IS-A relationship with the main article $P$. This feature corresponds to a soft instance classification of $P$.

Once these features are computed, they are added to the initial feature vector:

$$Φ ← [Φ; φ_{54}, φ_{55}] = [φ_1, φ_2, φ_3, ..., φ_{53}, φ_{54}, φ_{55}]$$

4 Learning Model and Experimental Evaluation

As described in Section 1.1, the distant positive examples are created from 3,021 list pages in Wikipedia. List pages have widely varying sizes, with a few very large list pages generating a significant number of distant positive pairs – about 30% of the list pages generate fewer than 7 positive pairs. Consequently, using all distant positive pairs for training is likely to introduce significant bias. In order to maintain diversity while also keeping training times feasible, we ran a series of evaluations in which the positive training examples were created by random sampling of $k$ distant positive pairs for each list page (or less, if the list was smaller).

We used the SVM$^{light}$ implementation of binary classification SVMs, with a quadratic kernel and the standard parameters. We evaluated the model in a 10-folds cross-validation scenario, in which training is done on 90% of the data and testing on the remaining fold of 10%, repeated for all 10 folds. The evaluation was done on datasets that were generated by varying the sampling parameter $k$ from 3 pairs to 7 pairs per list page. Table 4 below shows the performance of the binary classification model for the different values of the sampling parameter.

As expected, the performance increases slightly with $k$, both in terms of accuracy and $F_1$-measure. Since efficiency is also a concern, we decided to use $k = 5$ in our final evaluation on manually labeled data. Table 5 shows the performance of the model that was trained on the distant labeled data with $k = 5$ and tested on the manually labeled data, in two versions: with and without bootstrapping features.

The model that uses bootstrap features obtains increased performance in terms of both accuracy and $F_1$-measure. Figure 1 shows the precision vs. recall curve obtained by varying a threshold on the margin score output by the SVM model.

2svmlight.joachims.org

12
<table>
<thead>
<tr>
<th>$k$</th>
<th>Dataset Size</th>
<th>Accuracy</th>
<th>$F_1$-measure</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>13,936</td>
<td>77.8%</td>
<td>80.4%</td>
<td>87.1%</td>
<td>74.6%</td>
</tr>
<tr>
<td>4</td>
<td>18,477</td>
<td>77.8%</td>
<td>80.2%</td>
<td>87.0%</td>
<td>74.4%</td>
</tr>
<tr>
<td>5</td>
<td>22,794</td>
<td>78.2%</td>
<td>80.4%</td>
<td>87.1%</td>
<td>74.6%</td>
</tr>
<tr>
<td>6</td>
<td>26,938</td>
<td>78.2%</td>
<td>80.1%</td>
<td>87.1%</td>
<td>74.2%</td>
</tr>
<tr>
<td>7</td>
<td>30,176</td>
<td>78.5%</td>
<td>80.7%</td>
<td>87.4%</td>
<td>74.9%</td>
</tr>
</tbody>
</table>

Table 4: Model performance on the distant labeled dataset.

<table>
<thead>
<tr>
<th>Bootstrapping</th>
<th>Accuracy</th>
<th>$F_1$-measure</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>77.5%</td>
<td>80.8%</td>
<td>81.2%</td>
<td>80.3%</td>
</tr>
<tr>
<td>No</td>
<td>76.5%</td>
<td>80.2%</td>
<td>79.4%</td>
<td>81.0%</td>
</tr>
</tbody>
</table>

Table 5: Model performance on the manually labeled dataset.

![Recall vs. Precision curve](http://florida.cs.ohio.edu/wpgraphdb)

Figure 1: Recall vs. Precision curve.

## 5 Graph Database

We ran the SVM model that was trained on the distant labeled data on the entire Wikipedia category graph and stored the results into a Neo4j graph database. We only imported categories that had a main article and ignored administrative or maintenance categories and titles. All the titles and edges in the category graph were imported in the graph database as nodes and links, respectively. Each link corresponds to a pair of titles $\langle P, C \rangle$ and is associated with two types of attributes in the graph database:

1. A *label* attribute that stores the binary classification label assigned by the trained SVM model.

2. A *score* attribute that stores the margin computed by the SVM model. Users of the graph database can choose a threshold for the score level, corresponding to a particular recall vs. precision level in the graph shown in Figure 1.

Table [6] below shows the number of title nodes and positive vs. negative relations that were classified and imported in the graph database. The database will be made publicly available at [http://florida.cs.ohio.edu/wpgraphdb](http://florida.cs.ohio.edu/wpgraphdb).

[http://neo4j.com/](http://neo4j.com/)
Table 6: Graph database statistics.

<table>
<thead>
<tr>
<th></th>
<th>Title Nodes</th>
<th>Title Links (P, C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category Nodes</td>
<td>144,941</td>
<td></td>
</tr>
<tr>
<td>Article Nodes</td>
<td>2,158,309</td>
<td>Positive Links</td>
</tr>
<tr>
<td></td>
<td></td>
<td>718,880</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Negative Links</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2,511,464</td>
</tr>
<tr>
<td></td>
<td>2,303,250</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3,230,344</td>
</tr>
</tbody>
</table>

Acknowledgments

This work was supported in part by the National Science Foundation IIS awards #1018613 and #1018590, and an allocation of computing time from the Ohio Supercomputer Center.

References
