Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

Unsupervised Learning of an IS-A Taxonomy from a Limited Domain-Specific Corpus

Daniele Alfarone and **Jesse Davis**
Department of Computer Science, KU Leuven
Celestijnenlaan 200A - box 2402, 3001 Leuven, Belgium
{daniele.alfarone,jesse.davis}@cs.kuleuven.be

Abstract

Taxonomies hierarchically organize concepts in a domain. Building and maintaining them by hand is a tedious and time-consuming task. This paper proposes a novel, unsupervised algorithm for automatically learning an IS-A taxonomy from scratch by analyzing a given text corpus. Our approach is designed to deal with infrequently occurring concepts, so it can effectively induce taxonomies even from small corpora. Algorithmically, the approach makes two important contributions. First, it performs inference based on clustering and the distributional semantics, which can capture links among concepts never mentioned together. Second, it uses a novel graph-based algorithm to detect and remove incorrect *is-a* relations from a taxonomy. An empirical evaluation on five corpora demonstrates the utility of our proposed approach.

1 Introduction

Domain ontologies play an important role in many NLP tasks, such as Question Answering, Semantic Search, and Textual Entailment. Taxonomies are considered the backbone of ontologies, as they organize all domain concepts hierarchically through *is-a* relations, which enables sharing of information among related concepts.

Many handcrafted taxonomies have been built that capture both open-domain (e.g., WordNet) and domain-specific (e.g., MeSH, for the medical domain) knowledge. Yet, our knowledge is constantly evolving and expanding. Consequently, even domain-specific, handcrafted taxonomies inevitably lack coverage, and are expensive to keep up-to-date. This has motivated the interest in automatically learning taxonomies from text. Initially, systems focused on extending existing taxonomies [Widdows, 2003; Snow *et al.*, 2006]. Recently, there has been growing interest in automatically constructing entire taxonomies from scratch. Existing approaches learn taxonomies by analyzing either documents on the Web [Kozareva and Hovy, 2010; Wu *et al.*, 2012] or a combination of a domain-specific corpus and the Web [Velardi *et al.*, 2013; Yang, 2012].

This paper presents TAXIFY, a novel domain-independent, unsupervised approach that learns a taxonomy solely from a domain-specific corpus. This helps focus a learned taxonomy on the most important concepts in a specific corpus and minimizes the risk of including unrelated concepts extracted from irrelevant documents. TAXIFY learns accurate taxonomies that include infrequently observed concepts and relations that frequentist approaches typically discard [Kozareva and Hovy, 2010; Yang, 2012]. Specifically, instead of discarding single-source edges, TAXIFY propagates evidence from multi-source edges to single-source edges extracted from the same context.

Algorithmically, we make two important contributions. *First,* we use a clustering-based inference strategy that exploits distributional semantics to improve a taxonomy's coverage. In contrast, previous approaches improved coverage by either only considering pairwise concept similarities [Snow *et al.*, 2004] or performing purely syntactic inference [Velardi *et al.*, 2013]. *Second*, we propose a novel graph-based algorithm to detect and remove incorrect edges from a taxonomy in order to improve its precision. In contrast, existing pruning techniques [Kozareva and Hovy, 2010; Velardi *et al.*, 2013] attempt to maximize a taxonomy's connectivity, and only specifically search for incorrect edges in special cases (e.g., breaking a cycle). Furthermore, they typically assume that edges covered by multiple paths are more likely to be correct. However, we argue and show empirically that in certain cases removing edges that appear in many paths can significantly improve a learned taxonomy's precision.

Empirically, we compare TAXIFY to two state-of-the-art approaches, Kozareva and Hovy [2010] and Velardi *et al.* [2013], on five different domains. We find that TAXIFY outperforms them on the tested corpora. Additionally, an ablation study shows that (i) our clustering-based inference increases the number of correct edges by between 25.9% and 68.6%, and (ii) our pruning strategy increases precision by between 8.0% and 15.0% on average. Finally, TAXIFY's source code, data, and a demo are publicly available at http://dtai.cs.kuleuven.be/software/taxify.

2 Taxonomy Learning

We first introduce some basic terminology. A **concept** is an entity (either abstract or concrete) relevant for a certain domain, expressed as a simple noun (e.g., *dolphin*) or a noun phrase (e.g., *Siberian tiger*). When two concepts appear in an *is-a* relation (e.g., *mammal* $\rightarrow$ *fox*), we refer to the most-specific concept (e.g., *fox*) as the **subtype**, and to the broader one as the **supertype** (e.g., *mammal*).

Given a plain-text domain-specific corpus, TAXIFY learns an *is-a* taxonomy. The taxonomy is modeled as a directed acyclic graph $G = (C, E)$, where C is a set of vertices, each denoting a concept, and E is a set of edges, each denoting an *is-a* relationship. An edge $(x, y) \in E$, written as $x \rightarrow y$, denotes that subtype $y \in C$ has supertype $x \in C$. Using a graph instead of a tree allows a concept to have multiple supertypes, which reflects how humans classify objects.

At a high level, TAXIFY works in four phases. First, an initial set of *is-a* relations is identified, validated and added to the taxonomy. Second, coverage is increased through a clustering-based inference procedure. Third, precision is improved by identifying and discarding incorrect edges. Fourth, a confidence value is computed for each fact.

2.1 Constructing the initial taxonomy

TAXIFY builds the initial taxonomy as follows.

Identify seed set of *is-a* relations. To identify an initial set of *is-a* relations, TAXIFY applies the Hearst patterns [Hearst, 1992] shown in Table 1 to the corpus. However, Hearst patterns often overspecify *is-a* relations, by including generic words (e.g., "large") in a concept. As a handcrafted blacklist of modifiers may not generalize well across different corpora, TAXIFY computes a TF-IDF-like domain-specificity score for each word w:

$$ds(w) = \frac{f_{corpus}(w)}{f_{Eng}(w)} \cdot \frac{1}{\log n_{Eng}(w)} \qquad (1)$$

where the first part models the term frequency normalized by its overall frequency in English [Liu *et al.*, 2005], while the second part favors rare terms. $n_{Eng}(w)$ and $f_{Eng}(w)$ are the absolute and relative frequency of w in English as approximated by the Google Ngram frequency [Michel *et al.*, 2011].

Next, TAXIFY canonicalizes each concept by processing the concept modifiers from left to right until it encounters the first modifier w such that $ds(w) > \alpha_1$, where α_1 is a user-defined parameter. All modifiers before w are discarded from the concept. After canonicalizing both a subtype y and its supertype x, the edge $x \rightarrow y$ is added to G.

Increase connectivity and coverage by syntactic inference. TAXIFY identifies additional relationships to include in the taxonomy by performing syntactic inference on concepts containing modifiers (e.g., *whale shark*) as these represent a specialization of another concept. Concretely, whenever TAXIFY adds a multi-word concept to the taxonomy, it also adds its linguistic head as direct supertype (e.g., *shark* $\rightarrow$ *whale shark*). Then, TAXIFY expands the coverage of the taxonomy by, for each non-leaf concept x, scanning the corpus to (i) find all noun phrases (NPs) appearing at least twice whose head is x and (ii) add these NPs to the taxonomy as subtypes of x.

Improve precision by domain-specific filtering. Domain-specific filtering [Liu *et al.*, 2005; Velardi *et al.*, 2013] is one way to increase the precision of a taxonomy. Given a domain-specificity threshold α_2, TAXIFY removes all single-word concepts c from the taxonomy for which $ds(c) < \alpha_2$, where ds is computed by Equation (1).

x such as $\{y_i,\}$* {(or|and) y_n}
x including $\{y_i,\}$* {(or|and) y_n}
y_1 $\{, y_i\}$* and other x
y_1 $\{, y_i\}$* or other x

Table 1: Hearst patterns used by TAXIFY. Both *subtypes* (y_i) and their *supertype* (x) must be noun phrases (NPs).

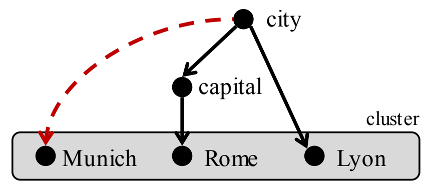

Figure 1: The solid edges are known *is-a* relations and the dotted line is an inferred one. By clustering together *Munich*, *Rome* and *Lyon*, TAXIFY infers that *Munich* is a *city* and not a *capital*.

2.2 Inferring novel facts

Since Hearst patterns occur infrequently, many interesting concepts will not be extracted. One way to improve coverage is to search for semantically-related concepts within the corpus. A well-studied solution is to exploit concepts that *co-occur* in lists [Cederberg and Widdows, 2003; Snow *et al.*, 2004; Davidov and Rappoport, 2006]. This approach suffers from two main drawbacks.

First, it can only capture links between concepts that explicitly co-occur. To overcome this limitation, TAXIFY adopts an approach based on distributional semantics [Harris, 1968]. To identify similar concepts, our approach extracts relations from the corpus. Two concepts are similar if they often appear in the same argument position of similar relations. For example, knowing that *cows chew grass* and *deer chew grass* provide some evidence that *cows* and *deer* are related concepts, as they share a common property. This allows TAXIFY to capture links between concepts that appear in separate sentences, or even documents.

Second, explicit co-occurrence does not necessarily imply that two concepts have the same immediate supertype. For example, knowing *capital* $\rightarrow$ *Rome* and that *Rome* is similar to *Munich*, may lead to the erroneous inference *capital* $\rightarrow$ *Munich*. To address this problem, TAXIFY clusters related concepts, searches for their most specific common ancestor, and assigns it as supertype for the new concepts. As illustrated in Figure 1, clustering *Munich* with *Rome* and the known concept *Lyon*, TAXIFY finds *city* as common ancestor, and thus infers that *Munich* is a *city*, and not a *capital*.

Computing pairwise similarity. TAXIFY runs a state-of-the-art OpenIE relation extractor [Fader *et al.*, 2011] on the entire corpus to obtain a list of triples of the form $r(c_{sub},\ c_{obj})$ (e.g., *chew(cow,grass)*), where $c_{sub}, c_{obj} \in$

C_{cand} are candidate concepts, and $r \in R$ is a relation. To reduce noise, TAXIFY removes from C_{cand} all concepts that appear in only one triple. Then, it creates a matrix $m_{feat} = |C_{cand}| \times 2|R|$ where:

- each row represents a concept $c \in C_{cand}$;
- each column is a pair (r, pos) with $pos \in \{subject, object\}$;
- each cell value (c, r_{pos}) is computed as $\log(1 + x)$, where x is the number of times c was extracted in position pos for relation r.

Each cell is then weighted by the negative entropy of the column vector that contains the cell, in order to reward features that are more discriminative in terms of a concept's semantics [Turney, 2005]. Intuitively, m_{feat} uses the list of triples where the concept appears as a set of features that capture contextual information about the concept.

Finally, TAXIFY creates a symmetric similarity matrix $m_{sim} = |C_{cand}| \times |C_{cand}|$ by defining the following pairwise concept similarity:

$$sim(c_1, c_2) = \left(\frac{\vec{c_1} \cdot \vec{c_2}}{\|\vec{c_1}\| \cdot \|\vec{c_2}\|} \right) \cdot \sqrt{\min(\|\vec{c_1}\|, \|\vec{c_2}\|)} \quad (2)$$

The first term is the cosine similarity between concepts $\vec{c_1}$ and $\vec{c_2}$ in m_{feat}. The second term weights the similarity based on the number of relations that a concept appears in, which gives higher weight to similarities backed by more evidence.

Clustering concepts. TAXIFY then infers new *is-a* relations in the following way. First, it uses K-Medoids to cluster the concepts based on the pairwise similarity values defined in m_{sim}. For each cluster that contains at least two known concepts (i.e., already in the taxonomy) and at least one new concept (i.e., found in a relational triple), it searches for the *lowest common ancestor* (LCA) among the known concepts (i.e., their most-specific common abstraction) and, if it exists, assigns it as the supertype to all new concepts in the cluster. Requiring each cluster to have two known concepts helps avoid incorrect inferences such as the one in our example with *Rome* and *Munich*. This procedure is repeated several times to minimize the impact of the initial random seed selection in K-Medoids. At the end of this iterative procedure, all edges inferred at least once are added to G.

Since this procedure needs to walk the graph, before executing it we break all present cycles.

2.3 Detecting incorrect edges

Learned taxonomies contain incorrect edges. TAXIFY attempts to detect incorrect edges in an unsupervised fashion by exploiting the following insight: when people provide illustrative examples of a subtype, they tend to use closely related supertypes [Kozareva and Hovy, 2010]. That is, we are more likely to write "*mammals such as bottlenose dolphins*" than "*organisms such as bottlenose dolphins*", even though both are true. Based on this observation, we postulate that it is unlikely for a taxonomy to contain a long path connecting x and y, if $x \rightarrow y$ was extracted by a Hearst pattern. If it does, it increases the chance that one of the edges in this long path is incorrect. Thus, the odds that an edge is incorrect increase each time it appears in such a path.

Figure 2 illustrates this on a portion of our learned animal taxonomy, where the red edge is the best candidate for exclusion from the taxonomy, as it appears in three long paths covered by three *is-a* relations extracted by a Hearst pattern, represented by dashed edges.

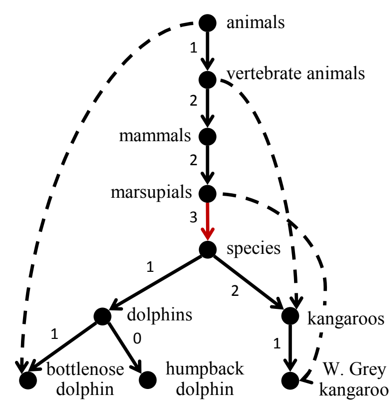

Figure 2: Part of our unpruned animal taxonomy. Dashed edges represent Hearst pattern extractions that cover long paths. The count associated with each edge represents the number of times that the edge is part of a path covered by a dashed edge. The red edge is incorrect.

Algorithm 1 outlines our procedure for detecting incorrect edges. As input, it receives the taxonomy G and a threshold β that discriminates between short and long paths. The procedure loops through the following steps until there is no edge covered by a long path. First, it counts how often each edge appears in a path longer than β that is covered by an edge extracted by a Hearst pattern. Second, it removes the highest-count edge from the taxonomy as well as all edges extracted by the same Hearst pattern in the same sentence, as their probabilities of being incorrect are correlated.

TAXIFY computes β for each corpus separately by using a standard outlier detection technique and sets $\beta = avg(L) + 2.5 \cdot std(L)$ [Healy, 1979], where L is a list of integers. L contains one integer for each edge $(x, y) \in E$ extracted by a Hearst pattern representing the length of the shortest path connecting x to y that does not include the edge itself.

Finally, since this procedure creates disconnected components in the graph, TAXIFY retains only the largest component, as the smaller ones typically diverge from the central topic of the corpus.

2.4 Assigning confidences to edges

Finally, TAXIFY assigns a confidence value to each edge, differentiating among edges: (i) directly extracted by a Hearst pattern, (ii) derived by syntactic inferences, and (iii) derived by clustering-based inferences. While not strictly necessary, confidence values provide richer output as they allow ranking edges by the strength of their evidence.

Algorithm 1: DETECTINCORRECTEDGES(G, β)

$m \leftarrow$ *empty map of < edge, counter >*
$E_{ext} \leftarrow getHearstExtractedEdges(E, G)$
foreach *edge* $e \in E_{ext}$ **do**
  $P \leftarrow getAllPathsCoveredBy(e, G)$
  foreach *path* $p \in P$ *s.t.* $length(p) > \beta$ **do**
    foreach *edge* $e_{covered} \in p$ **do**
      inc. counter in m at key $e_{covered}$

$e^* \leftarrow getHighestCountEdge(m)$
if *exists* e^* *with count* > 0 **then**
  remove e^* from G
  $E_{context} \leftarrow getContextualEdges(e^*)$
  remove all edges in $E_{context}$ from G
  DETECTINCORRECTEDGES(G, β)
else
  remove smaller components from G

Extracted edges. The initial confidence value of each extracted edge is:

$$p(e) = 1 - 0.5^{n_e^{(0)}}, \tag{3}$$

where $n_e^{(0)}$ is the number of times e was extracted by a Hearst pattern, as in NELL [Carlson *et al.*, 2010]. This models the fact that the more times an edge is extracted, the more likely it is to be correct. However, in smaller corpora many relevant relationships will only be extracted once,[1] thus a mere frequency count is unsatisfactory.

To motivate our solution, consider the following example. Assume that *mammal* → *deer*, *mammal* → *otter*, and *mammal* → *fox* are extracted by the same Hearst pattern from the same sentence. The first two edges are observed only once, while *mammal* → *fox* is extracted several times. Intuitively, the additional observations of *mammal* → *fox* give further evidence that the first two extractions are correct.

To capture this intuition, we iteratively update n_e by incorporating evidence from other edges as follows:

$$n_e^{(i)} = n_e^{(i-1)} + \sum_{e' \in Context(e)} p(e')^{i^2} \tag{4}$$

where $p(e')$ is given by Equation 3 and i is the iteration. In the first iteration, $Context(e)$ returns the set of edges extracted from the same sentence as e that have a higher confidence than e. In subsequent iterations, $Context(e)$ only returns those edges whose confidence became higher than e as an effect of the previous iteration. Since $p(e')^{i^2}$ is always less than 1, propagated evidence has always a weaker impact than a direct extraction. Furthermore, the effect of propagated evidence diminishes exponentially to model the intuition that each iteration increases the probability of introducing errors. The final confidence is obtained by using the final value of n_e instead of $n_e^{(0)}$ in Equation 3.

[1] 92% – 98% of all edges in the analyzed corpora.

As concrete example, suppose $n_e^{(0)}$(*mammal* → *deer*) $= 1$ and $n_e^{(0)}$(*mammal* → *fox*) $= 3$, and thus initially p(*mammal* → *deer*) $= 0.5$ and p(*mammal* → *fox*) $= 0.875$. Assume that in iteration 1 $Context$(*mammal* → *deer*) $= \{$*mammal* → *fox*$\}$, then $n_e^{(1)}$(*mammal* → *deer*) $= 1 + p$(*mammal* → *fox*) $= 1.875$. Assuming that in iteration 2 $Context$(*mammal* → *deer*) $= \emptyset$, then the final value for p(*mammal* → *deer*) $= 1 - 0.5^{1.875}$ which is approximately 0.727.

Syntactically-inferred edges. The confidence of a syntactic inference should be functional to the strength of the tie between a modifier and its head. For instance, we expect p(*shark* → *whale shark*) $> p$(*shark* → *aggressive shark*), as *aggressive* is a modifier that applies to more concepts and is thus less informative. To capture this intuition, TAXIFY computes a variant of the pointwise mutual information that accounts for statistical significance (PMI_{sig}) [Washtell and Markert, 2009] between the subtype modifier and its head:

$$PMI_{sig}(m; h) = \log \frac{f(m, h)}{\left(\frac{f(m) \cdot f(h)}{W}\right)} \cdot \sqrt{\min(f(m), f(h))} \tag{5}$$

where $f(m)$ and $f(h)$ are respectively the corpus frequencies of the modifier and the head, $f(m, h)$ is their joint frequency, and W is the total number of words in the corpus. To get the final confidence value for an edge, TAXIFY applies a log transformation to PMI_{sig} and then normalizes it to be in $[0, 0.8]$. This rescaling supports the intuition that an inferred edge should have a lower confidence than an edge observed several times in text because (i) if an extracted concept is incorrect, anything syntactically inferred from it would likely be incorrect too, (ii) even if an extracted concept is correct, a syntactic inference from it may be incorrect.

Edges inferred by clustering. Each edge $lca \rightarrow c$ inferred through clustering receives as confidence value:

$$p(lca \rightarrow c) = \max_i p_i(lca \rightarrow c) \tag{6}$$

where i ranges over the clustering iterations where $lca \rightarrow c$ was inferred, and p_i is defined as:

$$p_i(lca \rightarrow c) = \frac{1}{n} \sum_{k=1}^{n} P_{lca \Rightarrow c_k} \cdot sim(c_k, c) \tag{7}$$

where lca is the lowest common ancestor of the known concepts $\{c_1, \ldots, c_n\}$ that appear in the same cluster as c in iteration i, $sim(\cdot)$ is defined by Equation 2, and $P_{lca \Rightarrow c_k}$ is the product of the edge confidences on the path from lca to c_k. Intuitively, $p_i(lca \rightarrow c)$ is a similarity-weighted average over paths connecting each known concept c_k to lca. This is similar in spirit to how Snow *et al.* [2004] update an edge's confidence based on concepts that share a common ancestor in the graph.

As an example, we would calculate p(*city* → *Munich*) for the clustering shown in Figure 1 as:

$$p(\textit{city} \rightarrow \textit{M}) = \frac{1}{2}\Big(p(\textit{city} \rightarrow \textit{L}) \cdot sim(\textit{L}, \textit{M}) + p(\textit{capital} \rightarrow \textit{R})p(\textit{city} \rightarrow \textit{capital}) \cdot sim(\textit{R}, \textit{M})\Big)$$

where *R* stands for *Rome*, *L* for *Lyon*, and *M* for *Munich*.

3 Evaluation

The goal of the empirical evaluation is to address the following two questions:

1. How does our approach compare to state-of-the-art algorithms on this task?
2. What is the effect of each of the system's components on its overall performance?

Taxonomy evaluation is a hard task, as significantly different taxonomies can be equally correct in modeling a domain. Moreover, domains can be modeled at various levels of specificity. As a consequence, evaluation methods based on a comparison to a reference taxonomy (e.g., Zavitsanos, Paliouras, and Vouros 2011; Velardi et al. 2012) provide a meaningful comparison only when the set of concepts is fixed, as they penalize the introduction of new correct concepts. For these reasons, to fully compare systems that learn taxonomies from scratch we proceed in two steps. First, we manually inspect the learned taxonomies to assess the correctness of concepts and edges independently from their presence in a particular reference taxonomy. Second, given a reference taxonomy, we check how many of its concepts and edges are covered by the learned taxonomies.

To address the first question, we compare TAXIFY with two systems that can learn domain-specific taxonomies entirely from scratch, namely *K&H* [Kozareva and Hovy, 2010] and *OntoLearn* [Velardi *et al.*, 2013]. We implemented our own version of *K&H*. Since their bootstrapping approach terminates immediately in limited corpora, we relaxed some constraints to obtain a more meaningful comparison. Specifically, (i) we provided ten seed terms instead of one, (ii) we relaxed their patterns to match more than two subtypes, and (iii) we included basic syntactic inference. All comparisons with *OntoLearn* are made against their best run (*DAG*[0,99]), provided by its authors.

We use five real-world, plain-text corpora from different domains to evaluate and compare learned taxonomies, whose statistics are shown in Table 2. The first two are biomedical corpora, DDI[2] and PMC.[3] We also created three new corpora from Wikipedia, which we call ANIMALS, PLANTS and VEHICLES, by selecting all article abstracts that contain the words *animals*, *plants*, and *vehicles*, respectively.

To address the second question, we use a subset of these corpora to study the effect of each of TAXIFY's components on its overall performance.

3.1 Manual evaluation

Methodology. To assess the quality of the learned taxonomies, we report two evaluation metrics: *precision* and the *number of correct facts*. We use the number of correct facts or number of true positives instead of the recall because computing the recall requires a gold standard, which we lack.

Our taxonomies are too large to label each learned *is-a* relation manually. Therefore, we labelled 400 edges from each learned taxonomy to assess its accuracy. For *K&H* and *OntoLearn*, we randomly sampled and labelled 400 edges for each taxonomy. For TAXIFY, we divided all *is-a* relations into ten, equal-width bins based on their confidence (e.g., as done in Schoenmackers et al., 2010). Then we estimated a bin's precision and number of correct facts by labeling a random sample of 40 edges from the bin. An edge $x \rightarrow y$ is only labeled as correct if both (i) "y *is an* x" is a valid statement, and (ii) y and x are relevant concepts for the given domain. For example, the edge *tree* $\rightarrow$ *oak* would be labeled as incorrect when evaluating an animal taxonomy.

Since *K&H* requires root concepts as input, we picked *drug*, *medication*, *agent* for DDI, and *disease*, *gene*, *protein*, *cell* for PMC. We evaluated both the taxonomies rooted at these concepts and the unrooted taxonomies, since TAXIFY and *OntoLearn* do not need domain-specific inputs. For the Wikipedia corpora, we rooted all taxonomies at *animal*, *plant*, *vehicle* and their WordNet synonyms.

	#Documents	#Sentences per document	#Words
ANIMALS	25,016	10.5	5,656,301
PLANTS	64,545	7.7	9,370,609
VEHICLES	20,540	9.7	4,446,830
DDI	714	8.6	137,882
PMC	42,928	11.8	11,756,503

Table 2: Statistics on the corpora used in the experiments.

Parameter setting. We set TAXIFY's parameters as follows. For domain-specific filtering, we set $\alpha_1 = 0.4$ and $\alpha_2 = 1.7$ by assessing the accuracy of different filtering thresholds on approximately 500 manually labeled words on validation data from the PMC domain and an AI domain which was only used for algorithm development. α_2 is stricter than α_1 because many words are valid as concept modifiers (e.g., *white* shark), but not as a single-word concept (e.g., *white*). For the clustering-based inference, we set the number of clusters k for each corpus as the number of concepts to cluster divided by a parameter ψ. In each iteration, we randomly sampled ψ from the interval [3,7]. Randomizing ψ helps vary cluster compositions across iterations which improves the chance of finding an LCA. We did not try any other intervals for ψ.

Results. Figure 3 reports the results. In general, TAXIFY outperforms both competing systems. An exception is the large number of correct edges of *OntoLearn* on the unrooted DDI taxonomy, which comes at the cost of very low precision, mainly due to the extraction of several vague or generic concepts (e.g., *time*, *cause*, *rate*). Occasionally, *K&H* achieves precision comparable to TAXIFY, but totalling one order of magnitude fewer correct edges. *K&H*'s approach is indeed very conservative, as it is conceived for working at Web scale. Additionally, we found that TAXIFY's lower precision on PMC arises as a few incorrect edges link many irrelevant concepts to the taxonomy. TAXIFY's incorrect edge detection

[2] http://www.cs.york.ac.uk/semeval-2013/task9/

[3] http://www.ncbi.nlm.nih.gov/pmc/

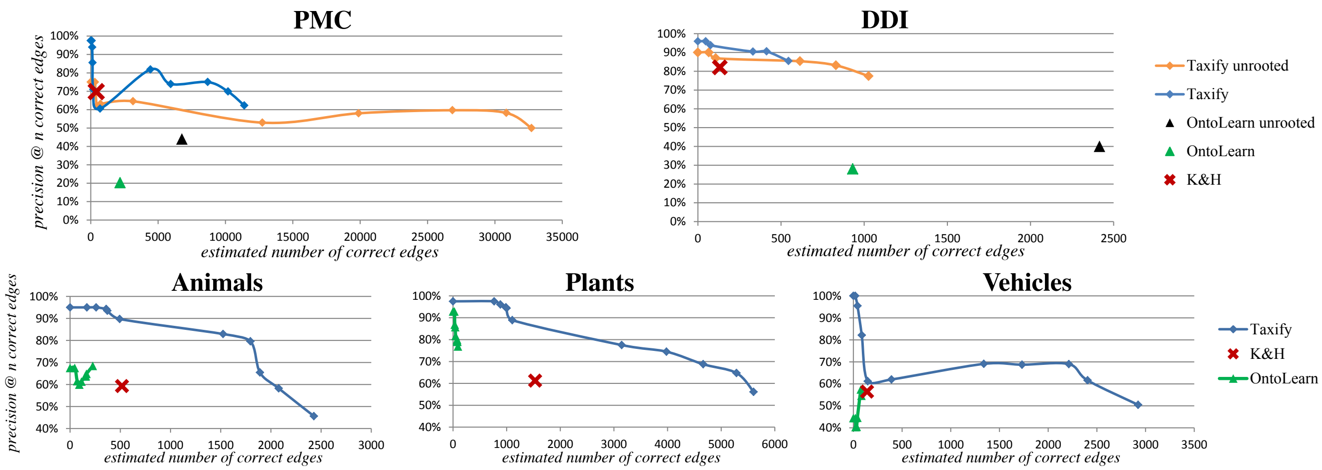

Figure 3: Comparison of TAXIFY with Kozareva and Hovy [2010] and Velardi *et al*. [2013]. Results are plotted as single points when edge's confidence scores are not available.

algorithm mitigates this problem, but it is slightly less effective due to the size and breadth of PMC.

3.2 Comparison with reference taxonomies

To assess the ability of taxonomy learners to cover reference concepts and edges, we compare the three taxonomies learned on the Wikipedia corpora to the corresponding WordNet subhierarchies rooted at *animal#n#1*, *plant#n#2*, and *vehicle#n#1*. We say that a learned concept c "covers" WordNet (i.e., $c \in C_{WN}$) if it appears in the target WordNet subhierarchy, while an edge $x \rightarrow y$ covers WordNet if x appears as a direct or indirect supertype of y in the subhierarchy. Table 3 shows that TAXIFY is able to cover a significantly higher number of concepts and edges compared to the competing systems on all three domains. However, the absolute numbers remain low, suggesting that larger corpora are needed to achieve a higher coverage of WordNet. We did not replicate Kozareva and Hovy [2010]'s WordNet experiment because it is not suitable for evaluating systems that learn taxonomies from scratch, as previously discussed [Navigli *et al.*, 2011; Velardi *et al.*, 2012; 2013].

Additionally, we compared the taxonomy learned from the DDI corpus to the handcrafted biomedical *MeSH* taxonomy.[4] Table 4 shows that out of 545 correct edges extracted by TAXIFY, 170 appear in the MeSH taxonomy. The remaining correct edges can be classified into those that *refine* MeSH, and those that *extend* it. We say that two edges $x \rightarrow x'$ and $x' \rightarrow y$ refine the MeSH taxonomy if the edge $x \rightarrow y$ appears in MeSH as x' increases the MeSH's level of detail. All other correct edges extend MeSH by either introducing a new concept or assigning an additional supertype to a known concept.

3.3 Evaluation of TAXIFY's subcomponents

We performed an ablation study on DDI, ANIMALS and VEHICLES to analyze the impact of TAXIFY's components on its overall performance. We compare the full TAXIFY system to three variants: TAXIFY without syntactic inference, TAXIFY without clustering-based inference and TAXIFY without pruning. The results are shown in Figure 4.

First, syntactic inference significantly increases the number of correct edges in the learned taxonomies. Syntactically-inferred edges play two important roles in taxonomy learning. One, they ensure the connectivity of the taxonomy. Two, they add additional concepts as leaf nodes in the taxonomy, by capturing relations that are infrequently observed in text.

Second, including the clustering-based inference results in more correct edges in all domains. Specifically, it increases the number of correct edges by 25.9% on ANIMALS, 68.6% on VEHICLES, and 27.0% on DDI compared to the no clustering-based inference baseline. The inference adds 2110, 2495, and 84 new concepts, respectively, to the taxonomies.

Third, TAXIFY's pruning strategy consistently improves precision across all confidence levels. Averaged over all confidence levels, the edge removal improves precision by 8.0% on ANIMALS and 15.0% on VEHICLES. On DDI the improvement (0.7%) is marginal, as the unpruned taxonomy is already highly precise. Averaged across the three corpora, 81% of the edges removed by our pruning strategy were incorrect.

Finally, we evaluated the benefit that clustering-based inference provides over pairwise inference. We created a variant of TAXIFY that replaces our clustering method with a pairwise approach that assigns each new concept the supertype of the most similar concept in the taxonomy. Compared to the pairwise strategy, our clustering approach increased the number of estimated correct edges by 23.8% on VEHICLES

	ANIMALS		PLANTS		VEHICLES	
	$\lvert C_{\text{WN}}\rvert$	$\lvert E_{\text{WN}}\rvert$	$\lvert C_{\text{WN}}\rvert$	$\lvert E_{\text{WN}}\rvert$	$\lvert C_{\text{WN}}\rvert$	$\lvert E_{\text{WN}}\rvert$
TAXIFY	**532**	**314**	**1016**	**612**	**120**	**89**
OntoLearn	180	144	81	30	63	52
K&H	275	182	468	211	63	16
WordNet totals	3999	4356	4487	4699	520	585

Table 3: Comparison of the three systems in terms of extracted concepts ($|C_{WN}|$) and edges ($|E_{WN}|$) that appear in the target WordNet subhierarchy.

[4]http://www.ncbi.nlm.nih.gov/mesh

Evaluation	Count	Examples
Already in MeSH	170 (31.2%)	*non-steroidal anti-inflammatory drugs* → *aspirin*, *psychotropic agents* → *tranquilizers*
Refines MeSH	118 (21.7%)	*highly protein-bound drugs* → *captopril*, *oral anticoagulants* → *warfarin*
Extends MeSH	257 (47.1%)	*drugs* → *Tiagabine*, *TNF-blocking agents* → *HUMIRA*

Table 4: Comparison of the 545 correct facts of our *drug* taxonomy against the MeSH taxonomy.

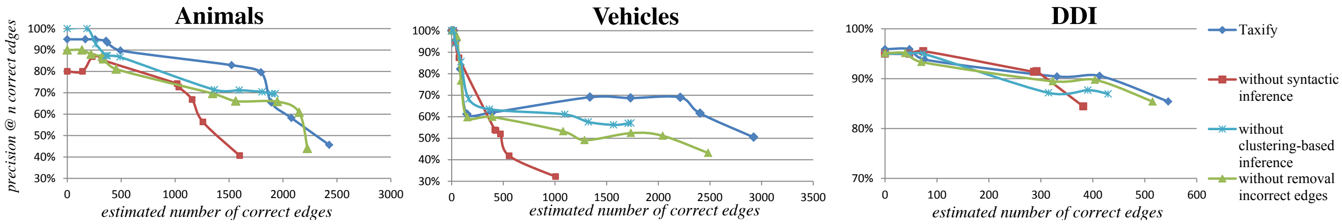

Figure 4: Results of the ablation study for TAXIFY.

and 11.9% on DDI, but resulted in a decrease of 3.2% on ANIMALS. The clustering approach also improved precision on average by 6.2% on ANIMALS, 11.7% on VEHICLES and 0.2% on DDI compared to pairwise inference.[5]

These results demonstrate that, in general, the full TAXIFY performs better than any of its variants. In particular, our two main innovations of using a clustering-based inference approach and our novel pruning strategy substantially contribute to TAXIFY's performance.

4 Related Work

The task of taxonomy learning can be divided into concept extraction and concept organization. While earlier systems focused on the first task only [Ravichandran and Hovy, 2002; Liu *et al.*, 2005], more recent efforts, like TAXIFY, tackle both tasks simultaneously.

Taxonomy learners typically aim to build either a general, open-domain taxonomy, or domain-specific taxonomies. These settings pose different challenges. Open-domain taxonomy learning (e.g., Ponzetto and Strube 2011; Wu et al. 2012) faces challenges such as analyzing very large textual corpora and coping with lexical ambiguity, but can leverage the massive redundancy in large corpora. Domain-specific taxonomies are typically induced from smaller corpora and thus cannot exploit high data density.

The most relevant related work are the domain-specific taxonomy learners [Kozareva and Hovy, 2010; Velardi *et al.*, 2013]. Kozareva and Hovy [2010] build an initial taxonomy by iteratively issuing Hearst-like patterns on the Web starting from seed terms, and then prune it to only retain the longest paths. Velardi *et al.* [2013] extract *is-a* relations from definition sentences using both a domain corpus and the Web. In our experiments, we found that definition sentences tend to extract generic supertypes, creating long chains of irrelevant concepts that limit the system's overall precision. Similar to Kozareva and Hovy, their pruning strategy optimizes the trade-off between retaining long paths and maximizing the connectivity of the traversed nodes. In contrast, TAXIFY can automatically discern if a long path is justified by the domain, or only caused by the presence of an incorrect edge, whose detection and removal significantly improves precision.

Regardless of the learning setting, several approaches have explored how to increase the coverage beyond Hearst pattern extractions, typically by searching for concepts similar to those already in the taxonomy. One line of work uses coordination patterns [Cederberg and Widdows, 2003; Snow *et al.*, 2004], which requires explicit concept co-occurrence. More recent work looks at distributional similarity (e.g., [Snow *et al.*, 2006]). However, these approaches require massive corpora. In contrast, TAXIFY can capture distributional similarity from a single smaller corpus. Furthermore, prior work (e.g., Cederberg and Widdows, 2003; Snow et al., 2004) tends to focus on similarities between *pairs* of concepts, whereas we show how a taxonomy learner can benefit from concept clustering.

5 Conclusions

We presented TAXIFY, an unsupervised approach for learning an *is-a* taxonomy from scratch from a domain-specific corpus. TAXIFY makes two key contributions. First, it uses an approach based on the distributional semantics and clustering to introduce additional edges in the taxonomy. Second, it proposes a novel mechanism for identifying and removing potentially incorrect edges. Empirically, these two contributions substantially improve the system's performance. Furthermore, we found that TAXIFY outperformed two state-of-the-art systems on five corpora. Our corpora varied in size from small to medium, and each setting provides different challenges. The smaller domains were more focused and dense, but had more implicit information whereas the larger domains were sparser and noisier, but had more explicit information. Our evaluation indicates that TAXIFY performs well in these settings, while Web-scale corpora (not tied to a specific domain) would pose different challenges and require a different approach.

In the future, we would like to integrate our system into tasks such as inference rule learning and question answering. Additionally, we would like to modify TAXIFY to work with

[5] Graphs for this experiment are available in the online supplement at http://dtai.cs.kuleuven.be/software/taxify

a combination of a domain-specific corpus and the Web.

Acknowledgements

We thank Steven Schockaert for his helpful feedback. This work was partially supported by the Research Fund KU Leuven (OT/11/051), EU FP7 Marie Curie Career Integration Grant (#294068) and FWO-Vlaanderen (G.0356.12).

References

[Carlson *et al.*, 2010] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. Toward an architecture for never-ending language learning. In *Proc. of the 24th AAAI*, 2010.

[Cederberg and Widdows, 2003] Scott Cederberg and Dominic Widdows. Using lsa and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In *Proc. of the 7th HLT-NAACL*, pages 111–118, 2003.

[Davidov and Rappoport, 2006] Dmitry Davidov and Ari Rappoport. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *Proc. of the 44th ACL*, pages 297–304. ACL, 2006.

[Fader *et al.*, 2011] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proc. of the 2011 EMNLP Conference*, pages 1535–1545, 2011.

[Harris, 1968] Z. S. Harris. *Mathematical structures of language*. Wiley, 1968.

[Healy, 1979] M Healy. Outliers in clinical chemistry quality-control schemes. *Clin. Chem.*, 25(5):675–677, 1979.

[Hearst, 1992] Marti A Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proc. of the 14th Conference on Computational Linguistics*, pages 539–545. ACL, 1992.

[Kozareva and Hovy, 2010] Zornitsa Kozareva and Eduard Hovy. A semi-supervised method to learn and construct taxonomies using the web. In *Proc. of the 2010 EMNLP Conference*, pages 1110–1118. ACL, 2010.

[Liu *et al.*, 2005] Tao Liu, XL Wang, Y Guan, ZM Xu, et al. Domain-specific term extraction and its application in text classification. In *8th Joint Conference on Information Sciences*, pages 1481–1484, 2005.

[Michel *et al.*, 2011] Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, et al. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, 2011.

[Navigli *et al.*, 2011] Roberto Navigli, Paola Velardi, and Stefano Faralli. A graph-based algorithm for inducing lexical taxonomies from scratch. In *Proc. of the 22nd International Joint Conference on Artificial Intelligence*, pages 1872–1877, 2011.

[Ponzetto and Strube, 2011] Simone Paolo Ponzetto and Michael Strube. Taxonomy induction based on a collaboratively built knowledge repository. *Artificial Intelligence*, 175(9):1737–1756, 2011.

[Ravichandran and Hovy, 2002] Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proc. of the 40th Annual Meeting of ACL*, pages 41–47. ACL, 2002.

[Schoenmackers *et al.*, 2010] Stefan Schoenmackers, Oren Etzioni, Daniel S Weld, and Jesse Davis. Learning first-order horn clauses from web text. In *Proc. of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1088–1098. ACL, 2010.

[Snow *et al.*, 2004] Rion Snow, Daniel Jurafsky, and Andrew Y Ng. Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems 17*, 2004.

[Snow *et al.*, 2006] Rion Snow, Daniel Jurafsky, and Andrew Y Ng. Semantic taxonomy induction from heterogenous evidence. In *Proc. of the 44th ACL*, pages 801–808. ACL, 2006.

[Turney, 2005] Peter D. Turney. Measuring semantic similarity by latent relational analysis. In *Proc. of the 19th International Joint Conference on Artificial Intelligence*, pages 1136–1141, 2005.

[Velardi *et al.*, 2012] Paola Velardi, Roberto Navigli, Stefano Faralli, and Juana Ruiz Martinez. A new method for evaluating automatically learned terminological taxonomies. In *Proc. of the 8th Conference on International Language Resources and Evaluation*, 2012.

[Velardi *et al.*, 2013] Paola Velardi, Stefano Faralli, and Roberto Navigli. OntoLearn Reloaded: A Graph-Based Algorithm for Taxonomy Induction. *Computational Linguistics*, 39(3):665–707, 2013.

[Washtell and Markert, 2009] Justin Washtell and Katja Markert. A comparison of windowless and window-based computational association measures as predictors of syntagmatic human associations. In *Proc. of the 2009 EMNLP Conference*, pages 628–637, 2009.

[Widdows, 2003] Dominic Widdows. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Proc. of the 2003 HLT-NAACL Conference*, pages 197–204, 2003.

[Wu *et al.*, 2012] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. Probase: A probabilistic taxonomy for text understanding. In *Proc. of the 2012 International Conference on Management of Data*, pages 481–492, 2012.

[Yang, 2012] Hui Yang. Constructing task-specific taxonomies for document collection browsing. In *Proc. of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1278–1289. ACL, 2012.

[Zavitsanos *et al.*, 2011] Elias Zavitsanos, Georgios Paliouras, and George A Vouros. Gold standard evaluation of ontology learning methods through ontology transformation and alignment. *IEEE Trans. on Knowledge and Data Engineering*, 23(11):1635–1648, 2011.