Learning to Integrate Relational Databases with Wikipedia

Doug Downey, Arun Ahuja
EECS Dept., Northwestern University
Evanston, IL 60208
{ddowney, arun.ahuja}@eecs.northwestern.edu

Mike Anderson
Rexonomy
Heidelberg, Germany
mrander@gmail.com

Abstract

Wikipedia is a general encyclopedia of unprecedented breadth and popularity. However, much of the Web's factual information still lies within relational databases, each focused on a specific topic. While many database entities are described by corresponding Wikipedia pages, in general this correspondence is unknown unless it has been manually specified. As a result, Web databases cannot leverage the relevant rich descriptions and interrelationships captured in Wikipedia, and Wikipedia readers miss the extensive coverage that a database typically provides on its specific topic.

In this paper, we present ETOW, a system that automatically integrates relational databases with Wikipedia. ETOW uses machine learning techniques to identify the correspondences between database entities and Wikipedia pages. In experiments with two distinct Web databases, we demonstrate that ETOW outperforms baseline techniques, reducing error overall by an average of 19%, and reducing false positive rate by 50%. In one experiment, ETOW is able to identify approximately 13,000 correct matches at a precision of 0.97. We also present evidence suggesting that ETOW can substantially improve the coverage and utility of both the relational databases and Wikipedia.

1 Introduction

Wikipedia is arguably the most comprehensive and frequently used knowledge base in existence. The Web-based encyclopedia contains user-contributed entries on a multitude of topics, providing detailed descriptions of millions of distinct entities and their interrelationships.

Nonetheless, it remains the case that much information on the Web resides in relational databases focused on a particular domain. For almost any conceivable topic, the Web contains a corresponding online database; examples include the USDA Nutrient Database for nutrition, the Internet Movie Database for films, and numerous similar databases focused on mountains, music, diseases, castles, digital cameras, and so on. For the most part, each database has coverage that—in its specific domain—greatly exceeds that of Wikipedia. However, because the databases are domain-specific, they lack useful connections to the more general knowledge found in Wikipedia.

In this paper, we present ETOW, a system that automatically *integrates* relational databases with Wikipedia by resolving precisely which Wikipedia page, if any, corresponds to each entity in a given relational database. This integration offers several benefits. For example, ETOW can enhance the relational database with helpful links into the general Wikipedia knowledge base. Likewise, as we illustrate, information from the database can be utilized to augment infoboxes on Wikipedia pages, or to create appropriate new pages. Further, in combination with recent automated techniques for categorizing Web pages in terms of Wikipedia concepts [Gabrilovich and Markovitch, 2007] and identifying mentions of Wikipedia concepts in text [Milne and Witten, 2008; Cucerzan, 2007], ETOW can link entities in relational databases to relevant content in the Web at large.

Resolving correspondences between database entities and Wikipedia pages is challenging primarily because multiple distinct entities may share the same name. For example, consider a "musicals" database containing a record for the Broadway hit "Chicago"; of the more than twenty Wikipedia pages corresponding to different meanings of the word "Chicago" (including a city, a typeface, a poem, a magazine, and so on), only one is a correct match for the musical. While Wikipedia does include a category system for articles, it is known to be both incomplete and unreliable [Wu and Weld, 2008]; further, even an improved category structure is unlikely to exactly match the relational structure employed in a particular database. Thus, identifying the correct page requires utilizing other clues as well, such as whether the Wikipedia page includes text indicative of the entity's type, or whether the page text mentions the attributes and relations of the entity in the database. ETOW employs machine learning techniques to effectively identify correspondences based on these features.

In this paper, we introduce the task of learning to automatically integrate relational databases with Wikipedia. Our contributions are as follows:

1. We present a general method, ETOW, which employs machine learning techniques to automatically resolve relational database entities to Wikipedia pages, using a small number of labeled examples per entity type.

2. In experiments with two distinct databases, we demonstrate that ETOW can effectively resolve thousands of entities to Wikipedia. ETOW is shown to achieve high precision (0.9) on average, at an acceptable level of recall (0.74). Compared with baseline algorithms, ETOW reduces error in terms of F1 score by 19% on average, and reduces false positive rate by 50%.

3. We present evidence suggesting that the integration performed by ETOW can offer substantial improvements to the coverage of both Wikipedia and the database.

The remainder of the paper is organized as follows. We define our task formally in Section 2, and present our system in Section 3. The experimental results are presented in Section 4, and we provide evidence suggesting ETOW's utility in applications in Section 5. Section 6 discusses related work, and the paper concludes with a discussion of future work.

2 Problem Definition

We consider a relational database consisting of a set of *entities* E, *relations* R, and *types* T. Each $r \in R$ is a binary relation over the set of entities.[1] Each entity is of exactly one *type* (analogous to a table in a database implementation). Each type defines a set of *attributes* which have numeric or symbolic values for each entity of the type.

For example, a nutrition database may contain a relation *is_rich_in* which holds between the entities *Dark Chocolate* and *Anti-oxidants*. Further, both *Dark Chocolate* and *Broccoli* may be members of the *Food* type in the database, characterized by attributes such as *Food.calories_per_serving*. Figure 1 shows an example from the company database used in our experiments.

We represent Wikipedia as a set P of pages. Each page $p \in P$ is described by attributes including its title, text, category information, and so on.

Our task is to resolve which Wikipedia page, if any, corresponds to each relational database entity. More formally, we say an entity *matches* a Wikipedia page if the concept described on the page is the same as that represented by the database entity. We then define our task as follows:

Definition 1 *The* **database-to-Wikipedia resolution problem** *is the task of finding a mapping* $\phi : E \rightarrow (P \cup \{\text{null}\})$ *from entities* E *in a given relational database to pages* P *in Wikipedia, such that* $\phi(e)$ *is a Wikipedia page matching the entity* e *if such a page exists, and* $\phi(e)$ *is* null *otherwise.*

Our task definition considers Wikipedia *pages* as the targets of entity resolution. This definition may be too narrow for cases in which database entities refer to concepts described on only a portion of a Wikipedia page (for example, "Dark Chocolate" is described on a portion of the "Chocolate" page). However, as we demonstrate in our experiments, the assumption that entities correspond to individual pages often holds in practice.

[1] Our discussion and experiments focus on binary relations; the extension to relations of higher arity is straightforward.

3 The ETOW system

ETOW, so-called because it maps entities to Wikipedia, solves the database-to-Wikipedia resolution task using machine learning. Starting with a small set of seed correspondences, ETOW trains a classifier for each type to estimate whether a given pair $(e, p) \in E \times P$ is a correct match. Below, we describe a set of simplifying assumptions ETOW makes for tractability, and then describe the ETOW algorithm, classifier, and feature set.

3.1 Assumptions

For a reasonably large relational database containing millions of entities, there are *trillions* of potential correspondences between the database entities and Wikipedia's millions of pages. Clearly, narrowing the space of possible matches is required. We employ the following simplifying assumptions:

- We assume that each entity e has a *name* attribute, such that if e matches a page p, then the name of e is the title of p, with the potential addition of disambiguating text in trailing parentheses (as is standard in Wikipedia to denote specific senses of a term, e.g. "Chicago (2002 film)").
- We assume each database entity matches at most one Wikipedia page.

These assumptions dramatically simplify the resolution task, and hold the vast majority of the time in practice. In our experiments, the first assumption reduced the number of matches ETOW considered by more than four orders of magnitude, and was in fact true for more than 95% of the entities. The percentage is so high partly because for entities referred to by multiple distinct names, Wikipedia typically includes redirect pages linking the multiple names to a single, unified page. For example, the page titled "William Henry Gates" redirects to a page titled with the more common name of the founder of Microsoft, "Bill Gates." This practice helps ensure that if an entity e is described on Wikipedia page p, either p or some page redirecting to p will be titled with the name for e employed in the database.

The second assumption held in all cases we examined. By eliminating many potential matches, this assumption improved precision in our experiments considerably.

3.2 Algorithm

The algorithm ETOW follows is shown in Figure 2. ETOW begins by applying the first assumption detailed above to obtain a set C of *candidate matches*: all pairs (e, p) such that the entity e and the page p refer to the same name. ETOW invokes a classifier (detailed below) that assigns a probability to each candidate match. For entities e for which some page p is a greater than τ likelihood match, ETOW applies the second assumption, choosing as the match for entity e the most probable page p according to the classifier. The use of a probabilistic classifier and threshold τ allows ETOW to trade-off precision and recall according to application requirements—we illustrate this capability of ETOW in our experiments.

3.3 Classifier and Feature Set

ETOW employs inductive learning to train the probabilistic classifiers it utilizes to identify matches. In our experiments,

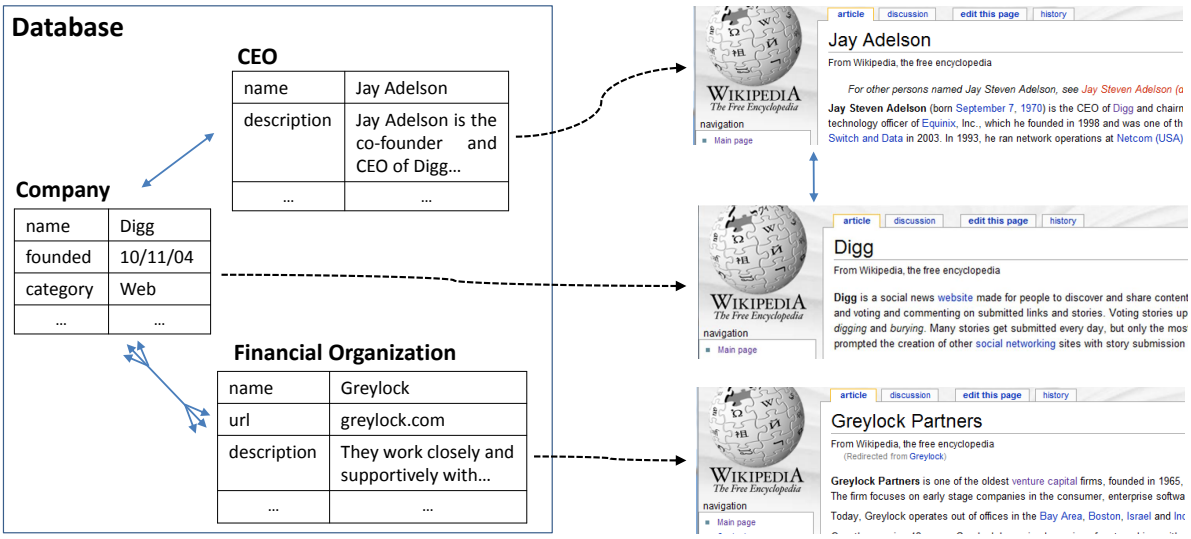

Figure 1: The database-to-Wikipedia entity resolution task. The goal is to obtain links between database entities of various types to their corresponding Wikipedia pages (indicated with dashed lines). Connections between two database entities indicate relationships between the entity types (multiple arrows signify a mapping to potentially many entities); connections between Wikipedia pages indicate hyperlinks.

ETOW(*Pages* P, *Entities* E, *Classifier* Φ_E)
  C = (e, p) such that $p \in P$ is titled with the name of e (modulo trailing parenthetical text)
  for $e \in E$:
    $\phi(e) := \arg\max_p \Phi_E((e, p) \in C)$
    if $\Phi_E(e, \phi(e)) < \tau$
      $\phi(e) :=$ null
  output ϕ

Figure 2: Pseudocode for ETOW at run-time. The classifier Φ_E assigns probabilities to candidate matches for entities in E, and the threshold τ is a parameter of the system.

we train a Support Vector Machine classifier for each entity type, using a small number of hand-labeled examples per type. We utilize the libSVM package, configured to produce probabilistic output [Chang and Lin, 2001].

ETOW's classifier estimates the probability that a given pair (e, p) is in fact a correct match. The features for this classification task were chosen based on two primary criteria. First, because ETOW is intended to be widely applicable, the features should be general-purpose and not tied to a specific database or domain. Second, as we wish train the classifier using only a small number of labeled examples, the feature space cannot be too large.

The features we employ for a given candidate match (e, p) are detailed below. Many of the features are computed based on "known" matches of similar types. In the standard ETOW algorithm, the known matches are simply those in the training set; however, as we describe in Section 4.3, the values can also be updated dynamically in an iterative, self-training configuration.

Entity Name/Page Title Features

Ambiguous entity names are disambiguated in Wikipedia page titles through the addition of text in trailing parentheses, as in "Chicago (2002 film)." Although the added text does not follow any consistent standard, it can be informative for identifying matches. For each pair (e, p), we created two features: a binary feature indicating whether p's title has text in trailing parentheses, and a continuous feature measuring the similarity between any parenthetical text of p to that of known matches of entities of e's type. We compute this similarity as the cosine measure between bag-of-words representations of the texts.

We would expect that candidate matches for more obscure or less ambiguous entity names are more likely to be correct. Thus, we include a feature giving the frequency of the entity name on the Web, as estimated from the Google n-grams data set,[2] as well as a feature giving the number of distinct Wikipedia pages titled with the entity name.

Textual Features

For correct matches (e, p), we expect the text of p to include some of e's attribute values or related entity names. Let a *related entity name* of an entity e be all names of entities e' where $r(e, e')$ or $r(e', e)$ occurs for some $r \in R$. We include a feature giving the cosine similarity between e's related entity names and a bag of words representing its attributes, a feature equal to the fraction of e's related entity names that appear in p, and a feature giving the Web frequency of the least-frequent related entity name of e found on p.

[2]For entity names longer than the five word limit of the data set, we estimate the frequency using a five-gram language model.

Relational Features

As shown in Figure 1, it may be the case that the relational structure of the database is reflected in the link structure of Wikipedia. Thus, we include features expressing how well the relational structure corresponds to Wikipedia links. Specifically, for each entity type related to e, we add a feature giving the fraction of matched entities e' related to e for which e''s match has a hyperlink to or from p.

Category Features

Wikipedia's pages are organized into a hierarchical category structure, where each page may be included in an arbitrary number of categories. Although the structure is inconsistent and incomplete, entities of a given type tend to be mapped to similar branches of the structure, generally speaking. We compute a "bag of categories" for each page p consisting of its categories and up to three parent categories. Our category feature is then the cosine similarity between the bag of categories for p and the bag of categories for all known matches of entities of e's type. For category features, we employ TF/IDF normalization in the cosine similarity computation.

Popularity Features

We expect measures of the *popularity* of the Wikipedia page p and the entity e to be informative for the classification of a candidate match. More popular database entities are more likely to have a corresponding Wikipedia page. Also, the popularity of an entity and its corresponding page should exhibit some correlation. As direct popularity information is not externally available, we use surrogate measures. For the media products database, we treat the *sales rank* attribute as a measure of an entity's popularity; for the companies database we use the string length of the database's content describing the entity. We approximate the popularity of a Wikipedia page p by the number of Wikipedia pages linking to p.

4 Experiments

In this section, we describe experiments measuring ETOW's effectiveness in the database-to-Wikipedia resolution task for two distinct databases. We begin by describing our data sets, and then present our results.

4.1 Data Sets and Experimental Setup

We experimented with two distinct relational databases. The first, *media products*, is derived from the Amazon.com product database. Our version of the database included three types: a *products* type consisting primarily of recordings and films; an *artist* type of contributors to the products (bands, composers, actors, directors, etc.); and a *track* type representing each recording's individual tracks. The second database, *companies*, is a subset of CrunchBase, an online database of information on companies and their funding sources. As illustrated in Figure 1, the three entity types in this database were *companies*, *CEOs*, and *financial organizations*.

The media products database consisted of about 961,000 products, 390,000 artists, and 7.8 million tracks. From this large database, we sampled a set of about 2,400 products which had at least one candidate Wikipedia page match. These products and their associated artists and tracks comprised a working set for the products database, which we employ in our experiments. All feature values for the media products experiments were computed relative to this working set of entities. The companies database consisted of about 15,000 company entities, 1,700 financial organizations, and 4,000 CEOs; we used this database in its entirety.

To obtain training and test data, we selected a set of 40 entities of each type from the media products database, and 80 entities of each type from the companies database, with the requirement that each entity have at least one candidate match in Wikipedia. We hand-labeled the resulting candidate matches, producing a data set of 720 labeled match candidates for the media products database, of which 67 were correct, and 346 match candidates for the companies database, of which 157 were correct. Thus, our experiments test ETOW on two data sets with very different characteristics, as seen in the fraction of correct candidate matches (0.09 for the media products database, vs. 0.45 for companies) and the degree of ambiguity (6 possible matches per entity on average for media products, vs. 1.44 for companies).[3]

In our experiments, we measure performance via five-fold cross-validation over the labeled data (thus, training sets are relatively small—32 or 64 hand-examined entities per type). We partition the candidate matches by database entity, meaning that the same database entity never appears in the training set and the test set at the same time.

We compare the performance of ETOW with two intuitive baselines. The first, *All Exact*, marks a candidate match as positive *iff* its title exactly matches the name of the entity (i.e., the Wikipedia page title has no trailing parenthetical text). The second, *All Unambiguous*, marks a candidate as positive *iff* it is an exact match *and* the entity name is unambiguous in Wikipedia (note that this baseline can still generate false positives, because frequently the database refers to an entity other than the one appearing in Wikipedia).

We used a Gaussian kernel for the SVM employed in ETOW, with parameters chosen via grid search and 2-fold cross-validation on the training set. The parameter τ is set to 0.5 (placing equal emphasis on false positives and false negatives) unless indicated otherwise.

4.2 Results

We first investigate how well ETOW performs in the database-to-Wikipedia resolution task relative to baseline techniques. The results of this experiment are shown in Table 1. We measure performance in terms of accuracy (the fraction of candidate matches correctly classified as positive or negative), precision (the fraction of positively classified candidates that are in fact correct), recall (the fraction of correct matches that are classified as positive) and F1 (the harmonic mean of precision and recall).

The results indicate that ETOW is substantially more effective than the baseline methods in both domains. In terms of F1, ETOW reduces error (deviation from 1.0) by 23% over the best performing baseline on the media products data, and

[3] On average, Wikipedia contains 1.05 distinct pages per concept name, so both databases exhibit above-average ambiguity.

	Media Products				Companies			
	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy
All Exact	0.543	**0.657**	0.595	0.918	0.702	**0.962**	0.812	0.798
All Unambiguous	0.722	0.582	0.645	0.941	0.816	0.904	0.858	0.864
ETOW	**0.930**	0.597	0.727	**0.958**	**0.864**	0.892	**0.878**	**0.887**
ETOW + self-training	0.911	0.612	**0.732**	**0.958**	0.842	0.917	**0.878**	0.884

Table 1: Performance of ETOW on the database-to-Wikipedia resolution task. ETOW outperforms the baselines by a substantial margin on both data sets, reducing error in terms of F1 by an average of 19%, and false positive rate by an average of 50%, when compared with the best performing baseline. Self-training has only a minor impact on performance.

by 14% for the companies data, for an average error reduction of 19%. On both data sets, neither baseline has a level of precision that is likely to be high enough for many data integration applications; ETOW improves on this substantially, reducing the false positive rate of the most precise baseline by an average of 50% across the two data sets.

As noted in Section 5, different applications may have very different requirements for precision and recall. The second question we investigate is whether ETOW can be used to cater output toward high-recall or high-precision performance, by manipulating the probabilistic threshold τ. As shown in Table 2, the precision of ETOW does in fact increase markedly if we increase τ to 0.95, at the cost of some recall. When we lower τ to 0.05, we see that recall increases greatly at the cost of some precision.

	Media Products		Companies	
	Precision	Recall	Precision	Recall
τ=0.95	0.969	0.463	0.895	0.108
τ=0.05	0.594	0.851	0.726	0.994

Table 2: Performance of ETOW when varying the classification threshold τ. The precision or recall of ETOW can be increased substantially as τ varies.

4.3 Enhancement to ETOW: self-training

Several of ETOW's features for a given candidate match become more informative when other matches are known. For example, the relational features for a candidate match (e, p) are only helpful when matches for entities related to e are known. Similar are the categorical and title-based features that compare aspects of p to other pages known to match to entities of e's type. This suggests a strategy of first identifying the easy-to-classify matches, and using these to inform the classification of the more difficult candidates. As an example, the entity *Catherine Zeta-Jones* is unambiguous and easy to match. We would like to leverage this easy match to help us find the correct Wikipedia page for more difficult-to-match entities related to Zeta-Jones, like the ambiguously-named 2002 film "Chicago." This strategy seems promising, because the correct matching page ("Chicago (2002 film)") is indeed linked with the Zeta-Jones page, whereas the page for the city of Chicago, for example, is not.

We incorporate this intuition in ETOW using a semi-supervised self-training approach. After training ETOW on the training set, we apply the system to the unlabeled candidate matches. Those candidate matches with probability greater than δ are added as positive training examples, and those with probability less than $1 - \delta$ are added as negative examples, where δ is a parameter of the system. We then re-compute the feature values using the new matches, and re-train the classifier on the new features and augmented training set. After repeating this process for k iterations, we measure performance on the test set.

The results of this enhancement are shown in the bottom row of Table 1, using values of $\delta = 0.95$ and $k = 10$.[4] Self-training does *not* provide substantial improvement, on average. Overall performance is essentially unchanged, with recall increasing somewhat and precision falling.

We believe the reasons self-training does not improve performance are two-fold. First, any benefits from exploiting similarities between the Wikipedia hyperlink structure and the relational database structure are to a large degree obviated by the textual features we employ: when a page links to a related entity page, the anchor text is typically the related entity's name. Second, the feature space is small enough that the original labeled training data is relatively representative, so the additional training examples produced by self-training are less beneficial. Exploring self-training with larger feature spaces is an item of future work.

5 Applications

In this section, we investigate ETOW's value in applications. We illustrate how the integration performed by ETOW can provide new capabilities for online databases, and improve the coverage of both the databases and Wikipedia.

ETOW offers a number of possibilities for enhancing online relational databases. By augmenting the relational database with links to Wikipedia pages, or by directly harvesting Wikipedia links or content, we could dramatically improve the generality of a database's content and search capabilities. Extrapolating from our experiments with the high-precision version of ETOW, we estimate that the system is able to correctly identify matches for 13,000 artist entities in the media products database, at precision of approximately 0.97. Further, we find that the matching Wikipedia pages for the entities contain many outgoing links, for an average of 56 per page. The linked pages cover a multitude of topics not found in the database, such as the artist's educational background and related artistic movements. The ability

[4] In previous experiments, adjusting the threshold or the number of iterations did not result in substantial changes in performance.

to search a database based on these general relationships—retrieving all recordings by artists linked to one's hometown, for example—would offer an enriched user experience.

"Infoboxes" on Wikipedia pages list relevant attribute/value pairs in an organized format. Recent efforts have attempted to increase the coverage of infoboxes automatically using text extraction [Wu *et al.*, 2008]. Could ETOW be employed for the same task? In measurements with the companies database, we find this approach holds remarkable promise. For CEOs, the majority of the Wikipedia pages (64%) do not contain infoboxes at all, and in each of these cases an infobox could be created containing at least one attribute from the database. For companies and financial organizations, infoboxes are more common, and contain between 6-7 attributes on average. Many infoboxes are missing particular attribute values, and we find that in the correspondences identified by ETOW, the database information can augment the infoboxes with 2.6 values for financial organizations on average, and 2.5 values for companies, for an average of a 40% improvement in coverage.

ETOW also detects database entities that are *not* currently found in Wikipedia (i.e., those with $\phi(e) = \text{null}$). In these cases, the database information can be used to generate high-quality "stub" pages. Based on a random sample of existing Wikipedia pages, we expect that the stub pages generated from database information would be at least as comprehensive as that of 60% of existing CEO pages, and 53% of company pages. To avoid creating duplicate pages, for this task we employ the high-recall version of ETOW (with $\tau = 0.05$); for the companies database, this approach can create thousands of stub pages with a duplicate rate of less than 5%.

Lastly, the infoboxes in the Wikipedia pages ETOW identifies as matches can improve the coverage of the database. For the companies database, the Wikipedia infoboxes contain multiple fields—e.g., net worth for CEOs, or revenue for companies—not found in the database. For the matches ETOW identified, we found an average of 1.9 values per infobox that could be added to the database.

6 Related Work

To our knowledge, this work is the first attempt to automatically integrate relational databases with Wikipedia. Recent work aimed at integrating Wikipedia with the Cyc ontology provides strategies for a disambiguation problem similar to ours [Medelyan and Legg, 2008]. However, that work targets the Cyc common-sense ontology, in contrast to our goal of a general architecture for integrating relational databases with Wikipedia.

Section 5 establishes the potential value of ETOW for automatically augmenting Wikipedia infoboxes. Another strategy for this a task is to extract information from text on the Web [Wu *et al.*, 2008]. Our work is complementary to this approach. ETOW augments Wikipedia using relational databases, which (even when online) are often not amenable to extraction methods that detect assertions in running text, like those employed in [Wu *et al.*, 2008]. Databases also offer higher precision than that of current extraction techniques.

Recent efforts to automatically construct a database from the information in Wikipedia infoboxes [Auer and Lehmann, 2007] suggests an alternative strategy for integrating databases with Wikipedia: first construct a database from Wikipedia infoboxes, and then apply well-studied methods of database integration (see e.g., [Doan and Halevy, 2005]). In contrast to this strategy, ETOW can be applied in the many cases in which no Wikipedia infobox is present. Lastly, in contrast to recent efforts toward linking mentions of concepts in text to their corresponding Wikipedia page [Milne and Witten, 2008; Cucerzan, 2007], our focus is on integrating relational databases, rather than textual content.

7 Conclusions and Future Work

In this paper, we presented ETOW, a general-purpose mechanism for integrating relational databases with Wikipedia. ETOW uses machine learning techniques to identify correspondences between database entities and Wikipedia pages. In experiments with two distinct databases, ETOW was shown to outperform baseline techniques.

ETOW and related research efforts present exciting possibilities for utilizing Wikipedia to perform large-scale semantic integration of online databases and Web content in general. In future work, we plan to experimentally investigate applications of ETOW and evaluate on additional data sets. We also plan to investigate active learning techniques, which offer the promise of improved accuracy while maintaining ETOW's limited need for human-annotated input.

References

[Auer and Lehmann, 2007] Sören Auer and Jens Lehmann. What have innsbruck and leipzig in common? extracting semantics from wiki content. In *Proc. of ESWC*, 2007.

[Chang and Lin, 2001] C. Chang and C. Lin. *LIBSVM: a library for support vector machines*, 2001.

[Cucerzan, 2007] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proc. of EMNLP*, 2007.

[Doan and Halevy, 2005] AnHai Doan and Alon Y. Halevy. Semantic-integration research in the database community. *AI Mag.*, 26(1):83–94, 2005.

[Gabrilovich and Markovitch, 2007] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proc. of IJCAI*, 2007.

[Medelyan and Legg, 2008] O. Medelyan and C. Legg. Integrating cyc and wikipedia: Folksonomy meets rigorously defined common-sense. In *Proc. of WIKIAI*, 2008.

[Milne and Witten, 2008] D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proc. of CIKM*, 2008.

[Wu and Weld, 2008] Fei Wu and Daniel S. Weld. Automatically refining the wikipedia infobox ontology. In *Proc. of WWW*, 2008.

[Wu *et al.*, 2008] Fei Wu, Raphael Hoffmann, and Daniel S. Weld. Information extraction from wikipedia: moving down the long tail. In *Proc. of KDD*, 2008.