Knowledge-Guided Schema Discovery from Semi-Structured Text

Thomas Lin
University of Washington
tlin@cs.washington.edu

Abstract

A significant amount of information is expressed as the semi-structured, non-grammatical text found in auction listings and classified advertisements. There would be value in automatically fitting this type of information to relational database schemas. Work has been conducted on automatically populating such a database, and automatic schema mapping is a large area of research, but the problem of automatically generating such a schema is relatively unaddressed. Schema Discovery is the recent research thrust that addresses this. In this paper, we introduce the TESS system for knowledge-guided schema discovery from semi-structured text. TESS performs term extraction over listing text and applies semantic reasoning to differentiate common terms into categories. This approach differs from existing schema discovery work in that it requires no hand-labeling or training examples, and it takes advantage of existing large databases of collected semantic knowledge.

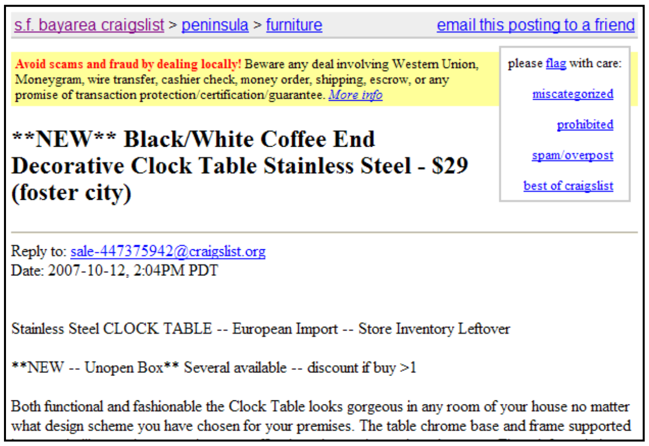
Problem Description

The goal of this project will be to extract database schemata from semi-structured text, specifically from auction and classified listings, which lend themselves well to a relational representation. The work is inspired by efforts going on in the field to automatically populate such a database in order to allow improved searching and some automated reasoning; in particular, Michelson & Knoblock [5] have created a tool named Phoebus, which tries to match cars listed on craigslist[1] with an entity table that knows what attributes should appear in any listing for a car, enabling intelligent querying. While Michelson & Knoblock mention the possibility of expanding the search domain automatically, their DB *schema* are actually created by hand, rather than discovered automatically. Such discovery would require, or at least be assisted by, some semantic understanding of the content of the target text. Tools that do large-scale knowledge representation, such as TEXTRUNNER [1], Cyc [3], and OpenMind Common Sense [7], may provide that background knowledge.

Motivation

The value of systems that automatically populate tables of semi-structured texts is obvious; they allow better searching over and machine access to such text. However, if such systems require manual creation of an appropriate database schema, they are substantially less flexible, as significant human effort is required to take advantage of them in even slightly novel domains. Combining the system we propose with a tool like Phoebus has the potential to provide many of the same benefits with substantially less human overhead.

We proposed to extract relational schemas rather than semi-structured (e.g., XML) schemas because relational schemas are the most widely used technique for describing structured data [2], and there are many good existing tools for querying and use of relational schemas.


s.f. bayarea craigslist > peninsula > furniture   email this posting to a friend

Avoid scams and fraud by dealing locally! Beware any deal involving Western Union, Moneygram, wire transfer, cashier check, money order, shipping, escrow, or any promise of transaction protection/certification/guarantee. More info

please flag with care:
miscategorized
prohibited
spam/overpost
best of craigslist

****NEW** Black/White Coffee End Decorative Clock Table Stainless Steel - $29 (foster city)**

Reply to: sale-447375942@craigslist.org
Date: 2007-10-12, 2:04PM PDT

Stainless Steel CLOCK TABLE -- European Import -- Store Inventory Leftover

NEW -- Unopen Box Several available -- discount if buy >1

Both functional and fashionable the Clock Table looks gorgeous in any room of your house no matter what design scheme you have chosen for your premises. The table chrome base and frame supported

Figure 1. Semi-structured online classified posting

Example

An example of furniture in online classifieds is shown in Figure 1. A human who read this post would be able to derive a schema appropriate for a database of this kind by using some background domain knowledge; for example, she may come up with the following fields: <u>Color</u>, <u>Price</u>, <u>Original Price</u>, <u>Material</u>, <u>Size</u>, <u>Usage</u>, <u>Condition</u>, and <u>Room Belongs In</u>.

Related Work

This work was initially motivated by work being done on automatically populating a database of classified listings given some existing schema, to which this work is an obvious complement. We also draw on existing work in

[1] http://www.craigslist.org

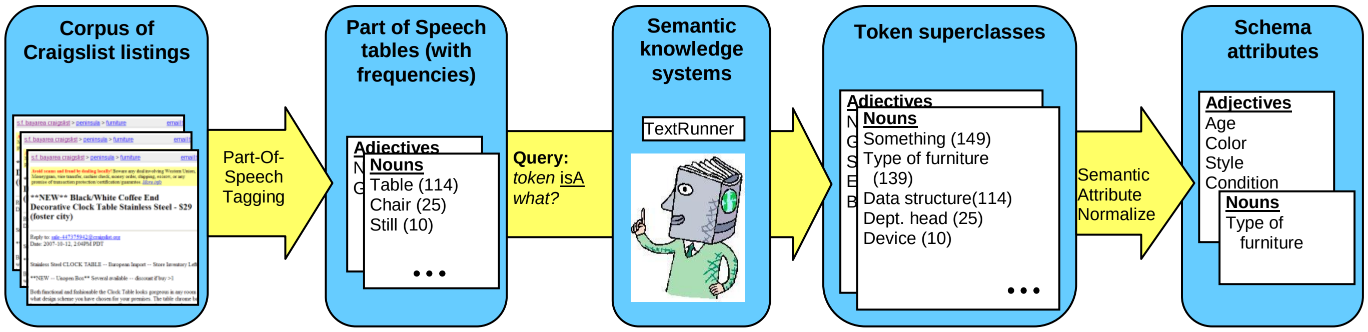


Figure 2. TESS Architecture

schema discovery in other domains [2], as well as relying on existing projects that provide semantic background [1,3,7]. Existing Schema Discovery projects like LIMBO [6] and Semi-CRF [4] differ in that they do not take advantage of semantic data, and tend to require training examples.

Relevant Technologies

We will rely on the following database systems and automatic database extraction systems:

- Phoebus: While we do not use the Phoebus engine itself, we capitalize on its approach at least conceptually;
- PostgreSQL: While this may seem obvious, a pre-existing full-fledged DBMS was critical.

Data Sources

Semantic input was provided by some combination of the following three very distinct technologies, which we had access to:

- TEXTRUNNER: Provides a very large amount of unstructured data in the form of tuples, such as "steel is-a metal";
- OpenMind: Provides several hundred thousand NL assertions entered by human volunteers.
- Cyc: Provides a smaller amount of high-quality, hand-crafted taxonomic and structural knowledge;

Just as one of the motivations for this work is the large pool of data available in a semi-structured form, one of the advantages is that this pool provided a large number of possible choices for a target.

Our initial work is with the craigslist online classifieds system, starting with a limited subcategory (furniture) and expanding to additional categories later.

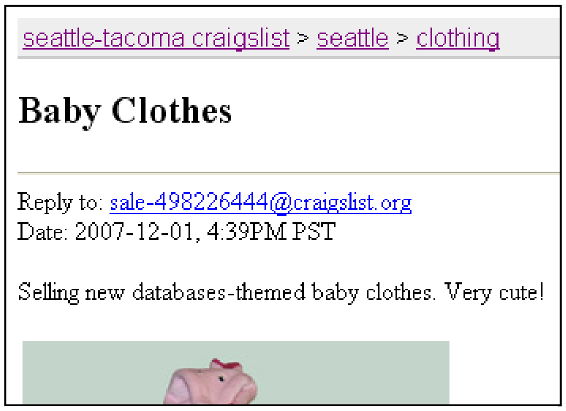
TESS

Overview

TESS generates schema attributes by using background domain knowledge like a human might. For example, if a human is reading craigslist car postings and reads about a red car, a blue car and a silver car, then the human might realize that all three postings described the "color" of a car, and realize that "color" is a relevant schema attribute for the cars domain. Similarly, TESS parses out the frequently occurring adjectives (e.g. "red," "blue," and "silver") and nouns from the postings, use knowledge sources to infer appropriate hypernyms (e.g. "color" is a hypernym of red, blue and silver), then filters the list of hypernyms into an appropriate set of schema attributes. Figure 2 shows the general TESS architecture.

Illustrative Example

To illustrate what each component of the architecture does, we will trace through a very short, one sentence document as an example. We will show how the document is transformed by each tool in sequence, and discuss the goal of each tool and the reason for using it.



seattle-tacoma craigslist > seattle > clothing

Baby Clothes

Reply to: sale-498226444@craigslist.org
Date: 2007-12-01, 4:39PM PST

Selling new databases-themed baby clothes. Very cute!

Figure 3. Example craigslist posting

Web Crawler

First, the TESS web crawler downloads postings from craigslist. For our example, the sample page that it downloads is shown in Figure 3. For that posting, the crawler extracts a "post" field of "Selling new databases-themed baby clothes. Very cute!" The crawler also saves information such as the title, date, location and post id number, but none of those are used by the current version of TESS. They are however worth storing because they may be used in future versions.

Part of Speech Tagging

The first thing that TESS does with each post is label the part of speech (e.g. noun, verb, adjective) for each word within the post. In our example, "Selling new databases-themed baby clothes. Very cute!" gets tagged into:

"Selling (verb) new (adjective) databases-themed (adjective) baby (noun) clothes (plural noun). Very (adverb) cute (adjective)!".

The reason for doing this part of speech tagging is so that we can better identify which specific terms within the post may be most useful for generating the schema attributes from. TESS uses the Stanford Natural Language Processing group's Part of Speech tagger [8] software for part of speech tagging.

Terms by Part of Speech

After the part of speech tagging, TESS organizes all of the parsed data into Part of Speech tables (the second blue box in the Figure 2 architecture diagram). Each table (one per part of speech) stores all terms which have that part of speech, as well as the number of times that word appears with that part of speech.

It does this in order to draw out groups of similar terms (in this case, groups like "nouns" and "adjectives," although the same concept applies to using shared syntax patterns to identify that words are likely to be similar). The frequency information at this point also allows people to see what the most frequently occurring nouns, adjectives, etc. in the original set are.

Our example sentence gets split into five tables, which are shown here as lists:

Verbs: "selling" (1 occurrence)

Adjectives: "new" (1 occurrence), "databases-themed" (1 occurrence), "cute" (1 occurrence)

Nouns: "baby" (1 occurrence)

Plural Nouns: "clothes" (1 occurrence)

Adverbs: "very" (1 occurrence)

In the current version, the main TESS system hypothesizes that the most useful information comes from nouns, proper nouns and adjectives, so only those tables are passed on for further processing. In future versions, we also plan to combine terms into noun phrases (e.g. "white chair" as a single entry, not just two separate entries) and cardinal phrases (e.g. "5 dollars" as a single entry).

Most Frequent Hypernyms

A hypernym is a word whose meaning denotes a superordinate or superclass. For example, "animal" is a hypernym of "bird," and "color" is a hypernym of "blue." This step uses hypernym information from a knowledge source to identify the hypernyms for each term in the adjectives, nouns and proper noun tables of the previous step. The reason we want hypernyms is because they are potentially useful as schema attributes. For example, if Steel, Glass, Metal, Wood, and Oak appear repeatedly in furniture postings, then semantic hypernym knowledge would suggest that their hypernym "material" may be a relevant schema attribute for furniture. The knowledge source which TESS uses is TEXTRUNNER.

TEXTRUNNER [1] is a project at the University of Washington for machine reading of the internet. TEXTRUNNER has read over 100 million web pages, and extracted several hundred million assertions from these web pages. TEXTRUNNER also has a model of which words are hypernyms for other words, based on statistics from Hearst Patterns. Hearst Patterns (e.g. "X, such as Y," "X, especially Y," "Y and other X") are sentence patterns that indicate suggest hypernym relationships. For example, if TEXTRUNNER observes many occurrences of statements like "Animals, such as cats …" and "… cats and other animals" in the web pages that it reads, then it would say that "animal" has a high probability of being a hypernym for "cat."

Going back to our example, we would query TEXTRUNNER for the hypernyms of all the adjectives, nouns, and proper nouns. In this case, those would be the adjectives "new," "databases-themed" and "cute," and the noun "baby." TEXTRUNNER would then return results, such as "new" can be a "condition" or "category," "cute" can be an "attribute," and "baby" can be a "child," "age range" or "event." The data is expressed as the following table:

word	hypernym	probability
new	condition	0.1
new	category	0.08
cute	attribute	0.11
baby	child	0.09
baby	age range	0.08
baby	event	0.07

Table 1. Example Hypernyms

Once TESS has the hypernyms and probabilities, it multiples the original frequencies by the hypernym probabilities to arrive at an ordering for all the hypernyms of adjectives, hypernyms of nouns, and hypernyms of proper nouns. In our example, we now have:

adjective hypernyms	sum(freq * probability)
attribute	0.11
condition	0.1
category	0.08

noun hypernyms	sum(freq * probability)
child	0.09
age range	0.08
event	0.07

Table 2. Hypernyms for Terms

Evaluating the tables, there are some potentially good schema attributes here (e.g. “condition” and “age range”), but there are also some hypernyms (e.g. “child”) that are meaningless as schema categories, and other hypernyms (e.g. “attribute” and “category”) which may be too vague to make good schema attributes.

Multiple words can have the same hypernym, so in those cases we would sum over their freq * probability values. When dealing with natural language, we run into the problem of word sense disambiguation. For example, the term “chair” has multiple word senses (meanings), so it could have hypernyms of “the head of a department” and “a piece of furniture.” The term “table” has multiple senses, so it could have hypernyms of “a database structure” and “a piece of furniture.” By taking the sum when hypernyms appear multiple times, we can identify that “a piece of furniture” scores higher than “the head of a department” or “a database structure” for the furniture category to overcome the word sense disambiguation problem.

Before deciding on TEXTRUNNER using Hearst Patterns, we explored a number of other options for hypernym information. One early method we tried was TEXTRUNNER using just “X is Y” relationships. However, this produced a high number of noisy results. We also looked at “X is Y” relationships in OpenMind Common Sense data. This produced fairly good data and clean results, but also gave us far fewer results because OpenMind only has on the order of 1 million entries while TextRunner has over 100 times that. We also tried taking superclasses from Cyc, but this ran into the problem of not having a method to easily map from craigslist terms into their equivalents in the Cyc language representation.

Attribute List

To address the problem of not all hypernyms being appropriate schema attributes, we decided to use the TEXTRUNNER hypernym system to generate a general, domain-independent list of which terms tend to be appropriate as schema categories. To do this, we queried the TEXTRUNNER hypernym system for all terms that have hypernyms such as “factor,” “measure,” “attribute” and “factor” (ideally terms that have all four of those as hypernyms). This gave us a probability-ordered list of nearly fifty thousand terms that could be appropriate as schema attributes. The top terms and their probability sums from this new “Attribute List” are time (0.70), age (0.63), price (0.59), weight (0.58), health (0.57), speed (0.57), color (0.56), ability (0.56) and language (0.56).

Another idea we looked at for identifying the best schema attributes was to consider the notion of generality level from Cyc. For example, the term “object” is too general while the term “University of Washington” is too specific. We thought that there would be a level of generality that would be ideal for schema attributes. This approach was promising, but the results were not as good as the Attribute List approach. As future work we may apply the generality level approach to the Attribute List to see if this produces an even better scoring for members of the Attribute List.

Schema Categories

We can now combine the list of adjective, noun and proper noun hypernyms with the Attribute List of appropriate schema attributes to identify what good schema attributes would be. The general algorithm for combining the lists is that we want terms which appear high on both lists. There are a variety of possible formulas that could accomplish this. Taking hypernym score as sum(freq * probability) from the hypernym tables, one formula would be *hypernym score * attribute list probability sum*. Another possible formula would be *hypernym score / (attribute list rank)*. However, this would weight too heavily in favor of the very low attribute ranks like 1, 2 and 3. Adding a constant value to the attribute rank would reduce this effect. The particular formula that TESS is using at the moment is:

score = hypernym score / (10 + attribute list rank)

If a term does not appear in the Attribute List at all, then it would receive a score of zero. This would then generate orderings for the best schema attributes based off the adjective hypernyms, noun hypernyms and proper noun hypernyms. Returning to our original example, if we had many posts (instead of just one), then our hypernym scores would be much higher, and we might arrive at tables like:

adjective schema	score
condition	25.43
attribute	11.04
category	8.32

noun schema	score

age range	37.53
event	18.34
child	0

Table 3. Example Schema Values

The user can take the top *n* terms to use as schema attributes. In this example, the top two terms "condition" and "age range" (bold in Table 3) would be fitting schema attributes for the clothing category that the post came from.

Solution Merits, Limitations and Properties

The TESS solution uses semantic knowledge to generate schema attributes. A key merit of this approach is that no human training or labeling of data is necessary, and this makes it more flexible. If data sets evolve over time, TESS can easily be run on them again. A limitation of the approach is that a large set of training examples are needed for the best results. For example, for this paper we analyze 4,000 craigslist postings per category. It is also worth noting that TESS also relies heavily on some of its components. The execution time is tied to factors like how long the part of speech tagger takes per posting, and the quality of results is influenced by the quality of the part of speech tagging and the quality of the hypernyms from the knowledge source.

Craigslist Furniture Results

Craigslist Furniture Results

The initial system focuses on data from craigslist furniture. We downloaded 4,000 craigslist furniture postings into a database. Then, we ran all of the postings through a part of speech tagger, and counted the number of occurrences of each type. There were 2,073 adjectives, 4,140 nouns, and 5,250 proper nouns.

The most popular adjectives, nouns, and proper nouns (with the frequency of each term) were:

Adjective	Noun	Proper Noun
new (708)	condition (1266)	Table (301)
great (596)	table (1199)	Queen (173)
good (555)	bed (601)	Set (150)
wide (450)	wood (579)	Mattress (146)
high (370)	room (441)	New (143)
deep (337)	chair (438)	ONLY (135)
old (321)	sale (406)	Antique (133)
available (304)	mattress (390)	OBO (119)
tall (286)	furniture (383)	Furniture (116)
interested (280)	piece (381)	CALL (115)
solid (279)	set (375)	Email (106)
nice (250)	top (369)	Wood (102)
other (238)	glass (357)	Oak (101)
excellent (234)	size (349)	AND (99)
free (218)	desk (329)	IKEA (92)
black (216)	call (326)	Chair (92)
beautiful (211)	sofa (324)	Bed (89)
white (191)	storage (312)	NEW (89)
comfortable (190)	home (297)	Great (89)
small (186)	frame (286)	Black (84)
perfect (172)	side (278)	Perfect (83)

Table 4. Craigslist Furniture Term Frequencies

Those results look fairly expected, as a standard craigslist furniture post would be like "Selling new wood table. Great condition!" and hit many of the top terms.

In terms of the direct hypernym superclasses, this is what we got for craigslist furniture:

Adjective	Noun	Proper Noun
word	place	place
way	way	way
term	factor	area
book	information	resource
organization	material	item
story	resource	organization
film	item	company
product	one	choice
band	area	book
understatement	problem	one
character	thing	thing
program	product	bit
movie	bit	product
system	element	term
album	book	day
site	object	word
option	issue	story
work	company	program
choice	choice	service
event	feature	film

Table 5. Craigslist Furniture Hypernyms

As we might expect, there are some terms (e.g. "word") that are too vague to be good schema attributes. Next, we normalize those results against the Attribute List to arrive at the following final list:

Adjective	Noun	Proper Noun
time	time	time
color	color	color
price	information	location
size	location	place
language	light	price
character	price	service
light	quality	quality
age	place	language

action system option location performance work organization change way day activity type	service exercise size activity cost factor area feature space unit weight way	light area type information size age activity feature action way day character

Table 6. Craigslist Furniture Schema Values

This list has some terms that do intuitively make good sense as schema attributes for craigslist furniture. For example, attributes like color, location, price, quality, size and weight all appear very relevant at first glance.

Evaluation

There are two aspects to the evaluation: how good the produced schema is, and how generalizable the schema-creation mechanism is.

Quality

We judge the *quality* of the schema by comparing it to a schema created by an independent craigslist furniture user who has no formal experience in schema creation. We also obtained a more standard furniture schema from the furniture web sites furnituretrader.com and usofficefurniture.net.

The first metric is to compare the Jaccard Index between the TESS-generated schema and the human-generated and standard schemas. The human-generated and standard schemas only contained 9 elements each, so only the first 9 schema attributes from TESS are used in this comparison.

Jaccard	Adjectives	Nouns	Proper Nouns
Human Generated	28.6%	28.6%	28.6%
Standard	20%	28.6%	28.6%

Table 7. Set Similarity to Evaluation Schema

On average there is a set similarity of 27-28% between the schema generated by TESS and the other two. It is also interesting to note that the results do not appear to vary much across different parts of speech. A slightly different metric is how many of the elements from the human and standard schema were present within the top 20 schema attributes returned by TESS. For this, we get:

% Contained in TESS top 20	Adjectives	Nouns	Proper Nouns
Human Generated	55.6%	55.6%	77.8%
Standard	66.7%	55.6%	77.8%

Table 8. Percent Containment of Evaluation Schema

These results are encouraging. TESS, without requiring any human labeled data is able to automatically generate the majority of the schema categories that a human would and that are used as the standard on furniture web sites.

Another metric for quality is an evaluation of how good/relevant the schema attributes produced by the system are, regardless of whether they match the human-generated or standard schemas. For this, we got:

Fraction good	Adjectives	Nouns	Proper Nouns
TESS Top 9	66.7%	77.8%	77.8%
TESS Top 20	60%	60%	65%

Table 9. Fraction of Results Good

These results show that most (74.1%) of the top results from TESS make for reasonable schema attributes, and also that better results tend to be ranked higher, as the Top 9 has a higher percentage of good results than the Top 20 does.

TESS currently forgoes simple techniques like syntactical pattern matching which might noticeably improve results in order to focus on what is achievable by semantic processing (which other systems have not studied) alone. It is hypothesized that TESS could be combined with other systems and techniques for even higher quality results.

Generality

It is important to test the *generality* of the system to make sure that the system is domain independent and no over-fitting took place to make it work better for furniture at the expense of other domains. To judge generality, we run the TESS system on the categories of craigslist cars and craigslist jewelry. For each of those categories, we had the crawler download 4,000 distinct posts and execute through to producing a schema. For craigslist cars, the resulting top 20 schema attributes per part of speech were:

Adjective	Noun	Proper Noun
time price exercise color	time price information color	time color service price

language	location	location
size	service	technology
character	size	size
work	quality	power
performance	energy	system
option	power	information
day	technology	energy
change	factor	quality
feature	system	feature
system	activity	type
location	feature	exercise
strength	day	place
activity	cost	environment
way	way	language
organization	condition	action
technology	action	activity

Table 10. Craigslist Cars Schema Values

For craigslist jewelry, the resulting top 20 schema attributes per part of speech were:

Adjective	Noun	Proper Noun
color	time	time
time	color	color
price	information	price
size	price	light
light	action	location
language	light	language
character	place	place
energy	location	information
age	style	service
culture	factor	type
performance	quality	energy
location	unit	size
quality	service	age
type	type	system
work	energy	style
option	exercise	quality
way	work	way
organization	day	unit
day	feature	area
change	weight	work

Table 11. Craigslist Jewelry Schema Values

The results do reflect the different domains, with the cars category having schema attributes like "technology," "power" and "energy" that are not present in furniture or jewelry, and furniture having schema attributes like "style" and "culture" that are not present in furniture or cars. A comparison of the TESS jewelry schema attributes to a schema generated by an independent user with expertise in jewelry indicated that, like with furniture, most of the schema attributes were covered within the top TESS–generated results. So, these results indicate that TESS is general enough that it can be applied to other domains, capture their interesting attributes, and give good results.

Future Work

As future work, there are several interesting ideas we may pursue for increasing the accuracy of the results generated by TESS. One direction, as mentioned earlier in the evaluation section, is to combine TESS with other systems and approaches that do not take a semantic approach, and see if they complement each other and what kind of improved performance can be achieved. For example, syntactical parsing is promising. A rule there could be that if a word (or sequence of words) commonly precedes a colon in the training set, then this could be a potential schema category. An example would be text like "price:" which is likely to appear in the training data and makes a good category.

Another direction is to refine the "part of speech tagging" aspect of the TESS architecture. For example, right now the term "5 dollars" gets parsed into two separate terms: "5" and "dollars." If the tagging could preserve these "cardinal value plus noun" constructs, then we could pass terms like "5 dollars" on to the TEXTRUNNER hypernym phase. A number of the human and standard schema attributes that TESS missed in the evaluation can conceivably be found with this method. An extension of this method would be to also consider full noun phrases as noun phrases instead of breaking them into their words.

We have some other ideas for how the general TESS system might be improved. For example, we chose to use the TEXTRUNNER hypernym system for hypernyms because of the coverage and quality. However, it's possible that systems like OpenMind and Cyc could be adapted to perform comparably or better for the specific subsets of data that they may specialize in. If this was the case, we could want to consider meta-algorithms that decide when to invoke each of the different systems. It may be worth looking at using TF*IDF to take advantage of the information we have about which post each word came from. Also, another component could be added after the schema values generation to pick the top *n* schema from the various part of speech tables that will maximize coverage and quality (e.g. we wouldn't want to recommend both "price" and "cost" as schema categories if they arise because they are redundant).

Lastly, a very interesting direction would be to combine TESS with the Phoebus system to see if a larger system can be created that can be pointed at arbitrary semi-structured sources, derive their schema, populate a database, and provide many of the other desirable features that Phoebus brings.

Conclusions and Contributions

The main idea presented by this paper is that general knowledge bases like TextRunner can help address the schema discovery problem in an intuitive way (similar to how humans might create schema categories) that requires no human data labeling, and that has not been explored by previous systems. Creating the TESS system was a valuable exercise in exploring the specific research challenges involved in such a system, and the evaluation results indicate that this approach is in fact effective and can generate good schema attributes.

TESS is valuable in that it is a working system that is able to operate effectively with noisy semi-structured real-world data. We hope to integrate TESS with systems like Phoebus in the future to create larger systems that can rapidly and automatically apply the advantages of databases to newly emerging semi-structured online data.

Acknowledgements

The author thanks Cynthia Matuszek for many important discussions and ideas on this topic. The author would also like to thank Magda Balazinska for valuable feedback on earlier versions of this work, and Mike Cafarella for suggesting the initial problem.

References

[1] Banko, M., Cafarella, M., Soderland, S., Broadhead, M., Etzioni, E. "Open Information Extraction from the Web." In *Proceedings of the 20th IJCAI*, India, 2007.

[2] Cafarella, M., Suciu, D., Etzioni, O. "Navigating Extracted Data with Schema Discovery." In *Proceedings of WebDB 10*, Beijing, China, 2007.

[3] Lenat, D. "Cyc: A Large-Scale Investment in Knowledge Infrastructure, *CACM* 38, no. 11. 1995.

[4] Mansuri, I. and Sarawagi, S. "A system for integrating unstructured data into relational databases." In *Proc. Of the 22nd IEEE International Conference on Data Engineering (ICDE)*, 2006.

[5] Michelson, M., Knoblock, C. "Phoebus: A System for Extracting and Integrating Data from Unstructured and Ungrammatical Sources." In *Proceedings Of AAAI 21*, Boston, MA 2006.

[6] Miller, R. J., and Andritsos, P. "Schema Discovery" *IEEE Data Eng. Bull.*, 26(3): 40-45, 2003.

[7] Singh, P., Lin, T., Mueller, E. T., Lim, G., Perkins, T., & Zhu, W. L. "Open Mind Common Sense: Knowledge acquisition from the general public." In *DOA/CoopIS/ODBASE*, 2002.

[8] Toutanova, K., Klein, D., Manning, C. and Singer, Y. "Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network." In *Proceedings of HLT-NAACL 2003*, pages 252-259.