Rapid Instance-Level Knowledge Acquisition for Google Maps from Class-Level Common Sense

Chris Welty, Lora Aroyo, Flip Korn, Sara M. McCarthy, Shubin Zhao
Google Research
{cawelty,l.m.aroyo,sara.m.mccarthy,shubin}@gmail.com, flip@google.com

Abstract

Successful knowledge graphs (KGs) solved the historical knowledge acquisition bottleneck by supplanting an expert focus with a simple, crowd-friendly one: KG nodes represent popular people, places, organizations, etc., and the graph arcs represent common sense relations like affiliations, locations, etc. Techniques for more general, categorical, KG curation do not seem to have made the same transition: the KG research community is still largely focused on methods that belie the common-sense characteristics of successful KGs.

In this paper, we propose a simple approach to acquiring and reasoning with *class-level attributes* from the crowd that represent broad common sense associations between categories. We pick a very real industrial-scale data set and problem: how to augment an existing knowledge graph of places and products with associations between them indicating the availability of the products at those places, which would enable a KG to provide answers to questions like, "Where can I buy milk nearby?" This problem has several practical challenges, not least of which is that only 30% of physical stores (i.e. brick & mortar stores) have a website, and fewer list their product inventory, leaving a large acquisition gap to be filled by methods other than information extraction (IE). Based on a KG-inspired intuition that a lot of the class-level pairs are part of people's general common sense, e.g. everyone knows grocery stores sell milk and don't sell asphalt, we acquired a mixture of instance- and class- level pairs (e.g. ⟨*Ajay Mittal Dairy*, milk⟩, ⟨GroceryStore, milk⟩, resp.) from a novel 3-tier crowdsourcing method, and demonstrate the scalability advantages of the class-level approach. Our results show that crowdsourced class-level knowledge can provide rapid scaling of knowledge acquisition in this and similar domains, as well as long-term value in the KG.

Introduction

From the outset, knowledge graphs (KGs) have prominently used crowdsourcing for knowledge acquisition, both from the perspective of scaling out graph creation and long-term maintenance, solving the historical knowledge acquisition bottleneck by revisiting the expert systems assumption that knowledge should be acquired from experts. As a result, popular KGs like Freebase (now Google's Knowledge Graph) are composed primarily of popular "common sense" entities and relations in the world that people are exposed to regularly and that can be acquired from and validated by the crowd.

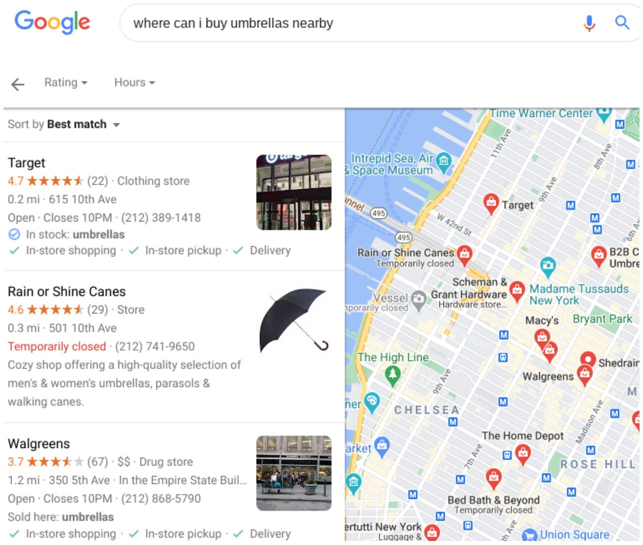
Similarly, today Google Maps overlays data on maps about the different places or establishments (stores, restaurants, hospitals, etc.) worldwide, and crowdsourcing plays a central role in the acquisition and maintenance of this information (Lagos, Ait-Mokhtar, and Calapodescu 2020). Users contribute opening hours, locations, reviews, etc., as well as categorical information about places such as whether it is a supermarket, department store, etc., which makes KGs a natural representation for this information.

Despite such heavy and widespread success of KGs for representing entities in the world and their properties, there has not been much attention paid in the research community to *class-level attributes* in KGs (Taylor 2017): graph edges between nodes that represent categorical terms, what they might mean and how to acquire them.[1] Practical and industrial KG edges remain almost exclusively at the instance level (e.g. McDonalds serves Big Mac), and a few KGs may encode class-level domain/range constraints (e.g. Restaurants serve Food), but no KG includes attributes of classes that represent our common-sense knowledge about them (e.g., Burger Joints serve burgers). There has certainly been a lot of research published in the sub-fields of Knowledge Representation on axiomatic knowledge acquisition (Ji et al. 2020), but these methods are not well-suited for crowdsourcing and have not made the transition to any industrial KG settings.

In this paper we explore the question of acquiring common sense class-level attributes from the crowd and applying those attributes effectively with other sources of information to solve a knowledge-base completion (KBC) problem (Bordes et al. 2013), in which we measure success by the precision and recall of graph edges. We take a particular problem, that of understanding the catalog of products sold at each store on earth. Such a KG could be used to answer questions like, "Where can I buy an umbrella nearby?" (see Fig. 1). We call this problem *local shopping* and it is one that is of interest tosearch engines like Google.[2]

[1] For the purposes of this paper we use the words *type, category, class* interchangeably, as well as *attribute, property, relation*.

[2] https://support.google.com/merchants/answer/9825611?hl=en



where can i buy umbrellas nearby

Target
4.7 ★★★★★ (22) · Clothing store
0.2 mi · 615 10th Ave
Open · Closes 10PM · (212) 389-1418
In stock: umbrellas
✓ In-store shopping · ✓ In-store pickup · ✓ Delivery

Rain or Shine Canes
4.6 ★★★★★ (29) · Store
0.3 mi · 501 10th Ave
Temporarily closed · (212) 741-9650
Cozy shop offering a high-quality selection of men's & women's umbrellas, parasols & walking canes.

Walgreens
3.7 ★★★★★ (67) · $$ · Drug store
1.2 mi · 350 5th Ave · In the Empire State Buil...
Open · Closes 10PM · (212) 868-5790
Sold here: umbrellas
✓ In-store shopping · ✓ In-store pickup · ✓ Delivery

Figure 1: Google Maps local shopping search results for umbrellas in NYC shows stores that sell them

Local shopping, compared to on-line shopping, poses a significant practical knowledge acquisition problem because shopping transactions do not occur on-line and therefore data about what products are being sold at what stores is not broadly available, making it a sort of "dark matter" of the web – we know it's there but can't directly observe it. Exceptions, such as Amazon Go, are rare, with less than 30% of stores worldwide having a website and even fewer that include a product catalog.[3] Indeed, our data shows that web pages and merchant feeds account for less than 1% of the total matrix of products at stores. To address this shortage of web information, we harness the crowd in three tiers: *users* around the world who have visited stores and voluntarily provide instance-level product availability (e.g. Ajay Mittal Dairy sells Milk); a much smaller set of *paid raters* who curate class-level attributes connecting common sense store and product categories (e.g. Grocery Stores sell Milk); and a very small set of *paid operators* who call stores to confirm the instance-level associations as evaluation ground-truth labels. The intuition behind this combination is that a lot of the instance-level associations are obviously true or false at the categorical level, and that acquiring knowledge at that level can jump-start the instance-level acquisition and help it be more productive: don't waste a user's efforts answering about milk or asphalt at an individual grocery store when simple common sense tells us the answer. Due to the prominence of common sense curation in our approach, we call the project *Shopping Sense*.

To our knowledge, acquiring class-level attributes from the crowd in order to jump-start a KBC problem has not been attempted before, and there are very few examples of KBC problems at this scale (tens of millions of stores wordwide and more than 10k products). The project and approach led to a successful worldwide launch of local shopping results overlaid on a map in a major search product, details of which will be provided upon acceptance to maintain anonymity. Due to the complexity and scope of the deployed project, we focus here on the real-world knowledge acquisition aspect of the work, and present a few simplified experiments that demonstrate how the acquired knowledge can be used for KBC. The contributions of this paper are primarily:

- To demonstrate that class-level bipartite knowledge acquisition can be effective in approximating instance-level knowledge (Error of Ratings section);
- A crowdsourcing approach to acquire such class-level knowledge for the local shopping problem (PRODCAT Data Collection Task section);
- Experimental results that show the effective combination of class- and instance- level knowledge from various sources used in the launched system (Results section).

The approach has generalized to other bipartite relations between places and types of entities that are organized in a taxonomy, such as dishes at restaurants, services at professional offices, etc., as well as a wide range of other bipartite graph problems where common sense or categorical knowledge prevails as defaults, such as ingredients for dishes, linnean taxonomies of living creatures, etc.

Problem Formulation

We start with an initial knowledge graph $\mathcal{G}'(\mathcal{I}_S \cup \mathcal{C}, \mathcal{R}_T \cup \mathcal{R}_{SC})$. The graph nodes are set of store and product categories $\{c_s \in \mathcal{C}_S\}$, $\{c_p \in \mathcal{C}_P\}$ respectively, so that $\mathcal{C} = \mathcal{C}_S \cup \mathcal{C}_P$ forms the set of all categories, as well as the store instances $\{i_s \in \mathcal{I}_S\}$ (we do not have access to product instances). The edges of the graph are the class/instance (also known as type) relation between store instances and store categories $\{\langle i_s, c_s \rangle \in \mathcal{R}_T\}$, and the subclass relation $\{\langle c_s, c'_s \rangle \in \mathcal{R}_{SC}\}$ the subclass relation with a disjointness constraint

$$\langle x, y \rangle \in \mathcal{R}_{SC} \implies \{x, y\} \subset \mathcal{C}_S \oplus \{x, y\} \subset \mathcal{C}_P$$

so that the relation is only defined over pairs of categories belonging to the same type. Lastly each of these primitive sets are disjoint $\mathcal{I}_S \cap \mathcal{C}_S = \mathcal{I}_S \cap \mathcal{C}_P = \mathcal{C}_S \cap \mathcal{C}_P = \emptyset$, making $\mathcal{G}'$ tripartite. As usual, $\mathcal{R}_{SC}$ forms a partial order within each (store and product) category partition, and is transitive over the subcategory relation so that $\langle x, y \rangle \in \mathcal{R}_T \wedge \langle y, z \rangle \in \mathcal{R}_{SC} \rightarrow \langle x, z \rangle \in \mathcal{R}_T$. This is meant to capture a traditional kind of knowledge-graph scenario.

Problem 1 *The* local shopping problem *is the extension of $\mathcal{G}'$ to $\mathcal{G}(\mathcal{I}_S \cup \mathcal{C}, \mathcal{R}_T \cup \mathcal{R}_{SC} \cup \mathcal{R}_I \cup \mathcal{R}_C)$ through the addition of the* class-level *product availability relation $\{\langle c_s, c_p \rangle \in \mathcal{R}_C\}$ and the* instance-level *product availability relation $\{\langle i_s, c_p \rangle \in \mathcal{R}_I\}$.*

The store instances $\{i_s \in \mathcal{I}_S\}$ represent individual physical stores like *Trader Joe's at 142 14th St.* (TJ142), each of which is typed with some number of store categories $\{c_s \in \mathcal{C}_S\}$ like *Supermarket*. The product categories $\{c_p \in \mathcal{C}_P\}$

[3] https://www.forbes.com/sites/jiawertz/2018/05/17/how-brick-and-mortar-stores-can-compete-with-e-commerce-giants/\#2019f5a23cc0

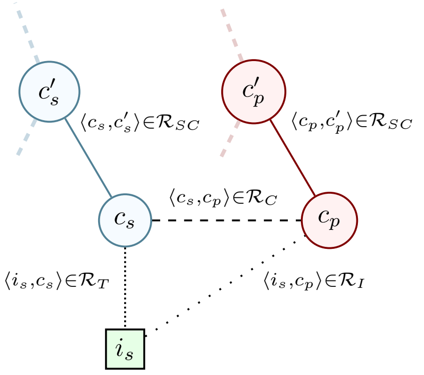

c'_s
$\langle c_s, c'_s \rangle \in \mathcal{R}_{SC}$
c'_p
$\langle c_p, c'_p \rangle \in \mathcal{R}_{SC}$
$\langle c_s, c_p \rangle \in \mathcal{R}_C$
c_s
c_p
$\langle i_s, c_s \rangle \in \mathcal{R}_T$
$\langle i_s, c_p \rangle \in \mathcal{R}_I$
i_s

Figure 2: Example subset of graph $\mathcal{G}$ with a store instance i_s, a store category c_s, its parent category c'_s, a product category c_p, its parent c'_p and the class- and instance- level product availability relations between them.

represent the types of products on the shelves of all stores, such as *Milk* or *Dairy*, so that $\{\langle TJ142, Milk \rangle \in \mathcal{R}_I\}$ means that particular Trader Joe's sells Milk. Note that a more complete definition of the local shopping problem would include the extension of $\mathcal{C}_P$ to instances (i.e. store inventory), but we do not have access to that data, and use this definition as a simplification that serves to answer most *local shopping* queries.

This simplification is best understood as a matrix $\mathbf{R} : \mathcal{I}_S \times \mathcal{C}_P$ representing $\mathcal{R}_I$, where $\mathbf{R}_{i,j}$ are observations (or predictions) that store i sells product j. With enough observed $\mathbf{R}_{i,j}$, collaborative filtering methods (e.g. matrix factorization) can be exploited to predict unobserved values from observed ones. Moving between matrix and graph representation can be done in a variety of ways, such as thresholding matrix values into discrete edges in $\mathcal{R}_I$, or using a graph formalism that supports confidence values on edges.

We argue that the real world grounding of the $\mathcal{R}_I$ association in people's everyday experience allows us to exploit meaningful common sense *categorical* knowledge for the problem of acquiring the edges in $\mathcal{R}_C$, and use simple defeasable methods to then infer the edges in the graph for the relation $\mathcal{R}_I$.

Vocabulary

The local shopping system and all the experiments described in this paper use the open *Google My Business* (GMB) categories[4,5] for store categories ($\mathcal{C}_\mathcal{S}$) and *Google Product Taxonomy*[6,7] for the product categories ($\mathcal{C}_\mathcal{P}$). Each set comes with a taxonomic structure that we encode as the R_{SC} relation, every category has at least one parent category with the exception of the top-level (most general) categories, and a few categories have multiple parents.

Terms	
KBC	Knowledge Base Completion
GMB	Google my Business (source store categories)
GPT	Google Product Taxonomy (source product categories)
UGC	User Generated Content
Crowd Sense	Our approach
Knowledge Graph	
$\{i_s \in \mathcal{I}_S\}$	set of store instances
$\{c_s \in \mathcal{C}_S\}$	set of store categories
$\{c_p \in \mathcal{C}_P\}$	set of product categories
$\langle c_s, c'_s \rangle \in \mathcal{R}_{SC}$	store subclass/class relation
$\langle c_p, c'_p \rangle \in \mathcal{R}_{SC}$	product subclass/class relation
$\langle i_s, c_s \rangle \in \mathcal{R}_T$	store class/instance type relation
$\langle c_s, c_p \rangle \in \mathcal{R}_C$	class-level product @ store availability relation
$\langle i_s, c_p \rangle \in \mathcal{R}_I$	instance-level product @ store availability relation
$\mathcal{G}'$	base KG of store/product classes and store instances
$\mathcal{G}$	$\mathcal{G}'$ extended with $\mathcal{R}_C$ and $\mathcal{R}_I$
$\mathbf{R}_{i,j}$	Likelihood that store instance i sells product class j
Crowd Task	
$w_{x,p}$	rater score for store (class or instance) x and product class p
$\alpha_{c,p}$	number of ”always” answers
$\nu_{c,p}$	number of ”never answers
$y_{i,p}$	number of ”yes” answers
$n_{i,p}$	number of ”no” answers

Table 1: Glossary of Terms

There are roughly 15k categories in $\mathcal{C}_\mathcal{P}$, that are similar in semantics to UPC codes, grounding out in 18 top-level categories. The GMB categories include many that are unrelated to local shopping, so we restrict ($\mathcal{C}_\mathcal{S}$) to those below *store*, resulting in roughly 1k with a single root.

These two taxonomies have different graphical structure: the product taxonomy is fairly deep, and the place taxonomy under store is fairly shallow, yet they align surprisingly well. For example, there is a deep taxonomy of products under "Grocery", and a store category "Grocery Store". There are a few misalignments, for example "Batteries" are under "Electronics" but are sold at "Drugstores". A few of these misalignments are ameliorated by hybrid categories like "Household products," which is an additional ancestor for "Batteries". Note that we do not change the taxonomies or memberships; as defined in Sec. Problem Formulation, we treat the initial graph $\mathcal{G}'$ as given.

Finally, Google Maps has tens of millions of stores worldwide that form $\mathcal{I}_\mathcal{S}$; each has a category label which is displayed in the maps UI under the place name and user rating, giving us the edges in $\mathcal{R}_\mathcal{T}$. A large part of these labels are assigned by merchants, some by users, some by operators and others by machine automation. These labels are generally high quality, with precision over 0.8. The largest source of inaccuracies are store labels that are more general than they need to be, when a more appropriate category exists. The labeling infrastructure requires a single "primary" category, while many places could be categorized in several ways.

Answers from Users

The system for which we performed the crowdsourcing described in this paper is quite large and complex, and is launched and available to users worldwide. It uses a DNN model to predict $\mathcal{R}_I$ pairs from signals that include information extraction (IE) from store web pages, direct merchant

[4] https://support.google.com/business/answer/3038177/\# categories

[5] https://bayareawebsitedesigner.com/gmb-categories/

[6] https://www.google.com/basepages/producttype/taxonomy.en-US.txt

[7] https://feedonomics.com/google_shopping_categories.html

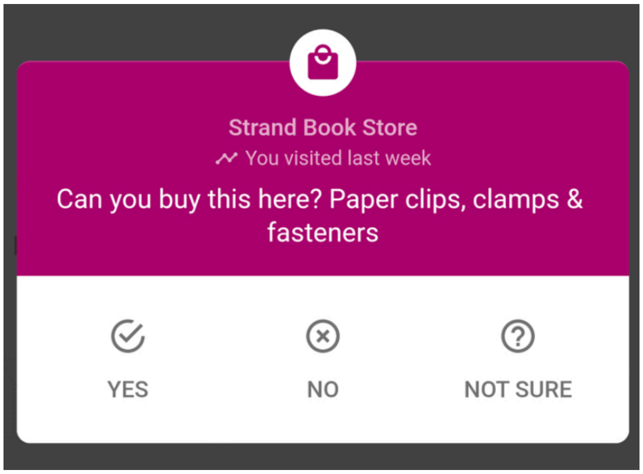
feeds, store type, and dozens of other features that include a significant amount of user-generated content (UGC).

Google Maps provides the facility to contribute and verify UGC of various kinds. Users voluntarily add reviews, photos, venue categorization, and attributes (e.g. “has Wi-Fi”) through these map-based apps. We have added a similar set of UGC yes/no questions for product availability (an example is shown in Fig.3). Volunteers who have visited the store and use our app can answer up to ten questions that provide us, after more than two years, with the largest source of instance-level pairs ($\mathcal{R}_I$), far greater than web IE, merchant feeds, or any other source. This UGC is in contrast and in addition to the class-level acquisition ($\mathcal{R}_C$) described in the next section.

The biggest challenge in collecting these UGC pairs is selecting ten products to ask at each store from the thousands of products we know about. We combine two methods to help target an optimal set of products per store: active learning and the categorical data we describe below. The full details and evaluation of this collection method is the subject of a future paper, we briefly point out that the $\mathcal{R}_C$ pairs we acquire from a paid crowd can serve as a guide for recognizing obvious $\langle i_s, c_p \rangle$ pairs, e.g. $\langle GroceryStore, Milk \rangle$ is obviously true, $\langle GroceryStore, Asphalt \rangle$ is obviously false, so if $\langle i_s, GroceryStore \rangle \in \mathcal{R}_T$, remove $Milk, Asphalt$ from the set of UGC targets for store i_s. Next, since the underlying launched system uses a large DNN model, we can rank the reduced set of products per store by their utility to the model, such as their distance to the classifier boundary.

Crowd Sense

The obvious way to gather the edges in $\mathcal{R}_I$ would be to use store inventory or transaction records. The problem with this approach is that *local shopping* is still mostly an off-line process worldwide, and we do not have large-scale access to transactional data that gives us these observations. Google provides stores a way to share their catalogs or inventory on-line, but much fewer than 5% of stores worldwide have made use of it. Our data showed that web pages and merchant feeds together accounted for less than 1% of the space of $\mathcal{R}_I$. To address the rapid acquisition and scaling of data in $\mathcal{R}_I$, we explored the acquisition of edges in $\mathcal{R}_C$.


Strand Book Store

You visited last week

Can you buy this here? Paper clips, clamps & fasteners

Figure 3: Example question used to gather UGC.

Crowd Hypothesis

The intuition driving our approach is that the crowd can provide the class-level knowledge ($\mathcal{R}_C$) by appealing to their common sense experience; everybody knows that, e.g. “All supermarkets sell milk”. Reality is more complicated, and the true problem is first dominated by what products are obviously *not* sold, and second by products that are *usually*, but not always, sold at some type of store. For example, Impact Wrenches are not ever found in Drugstores, and Wasabi Peas, while they are found almost exclusively in grocery stores, are not found in all of them. What we really aim for the crowd to provide is a distribution of products available at stores of a given type. This is where a lot of existing knowledge graph methods fail, especially at the class-level, as they rely on an assumption of discreteness.

It may seem that we could ask individual people to answer a question like, “What percent of stores of type c_s sell product c_p?” However, research in human computation has shown that individuals cannot reliably answer such questions (Surowiecki 2005). Based on previous work (Aroyo and Welty 2014, 2015; Welty et al. 2012), we hypothesized:

Hypothesis 1 *asking multiple raters about the same categorical pairs would produce a distribution of answers that approximate the real world distribution of* $\mathcal{R}_I$.

In other words, if 80% of raters say that oat milk is sold at grocery stores, then 80% of grocery stores will sell oat milk.

Before testing our hypothesis, we ran numerous pilots to tune task hyper-parameters, asking raters questions about 11k $\langle c_s, c_p \rangle$ pairs: from 5-25 raters per pair, 154 store types and 3600 products in five countries, and variations on the question phrasing. We settled on: five raters per pair, randomly selected from a pool of 130 raters in five countries, sourced from contracted operators through an in-house crowdsourcing platform, and the question, “Would you expect to find c_p products in stores of the category c_s?” with four answer options (“Always Available”, “Sometimes Available”, “Never Available”, “I don’t know”).

Under these settings, our final PRODCAT task (see below) gathered 25k $\langle c_s, c_p \rangle$ pairs with 5 labels per country, that through inference (*q.v.* Sec. Data Sources) resulted in over a billion $\langle i_s, c_p \rangle$ pairs, 99% of which were negative. It took six weeks to run and analyze the pilots, and two weeks to run the final task.

Data Collection Tasks

Another way to state our hypothesis is that the categorical crowd disagreement should reflect the real world distribution, but disagreement can have many causes that are not related to the desired distribution. The various pilot tasks we ran represented a gradual refinement of the data and task descriptions to eliminate disagreement from other causes. We report here on four different approaches:

RANDOM To confirm the sparsity of $\mathcal{R}_C$, we randomly and independently selected category pairs from $\mathcal{C}_S \times \mathcal{C}_P$, weighing the selection from $\mathcal{C}_S$ proportionally to the number of stores belonging to each category (i.e. larger categories are more likely to be selected). Pairs were presented to 5 raters from the same country. This RANDOM task confirmed that the vast majority of pairs are "obvious" negatives (asphalt at grocery stores, cars at violin shops, etc.), as more than 95% of the pairs resulted in 5 "Never" ratings.

SINGLETON To address the sparsity shown in RANDOM, we leveraged web signals (see Sec. Data Sources) to select pairs with more likelihood to be available at stores within a given category, and presented one pair at a time to 5 raters from the same country. This resulted in rating scores ranging from all-5 "Always Available" to all-5 "Never Available", but skewing towards the positive side. The SINGLETON task results showed disagreement from other causes, described in Sec. Ambiguity.

MATRIX To address the disagreement due to ambiguity (Sec. Ambiguity), we designed a novel matrix presentation of class-level pairs, with four $\{c_s \in \mathcal{C}_S\}$ as the columns and a set of 100-200 $\{c_p \in \mathcal{C}_P\}$ as the rows, depending on our ability to match products to the store categories using web signals. Figure 5 shows the matrix presentation (with data sampled through the PRODCAT method below). The advantage of this presentation is that raters familiarized themselves with a category and answered many questions related to it, rather than having to understand one pair at a time. This approach still produced some unwanted disagreements due to difficulty understanding some of the products, esp. very specific ones, and we were concerned that the web signals were biasing our sample towards availability patterns of online stores, rather than our target class of stores without web pages. Most importantly, the amount of time the raters spent per $\langle c_s, c_p \rangle$ dropped by 50%.

PRODCAT The final crowdsourcing task used the MATRIX presentation but changed to a dynamic method that sampled the $\langle c_s, c_p \rangle$ pairs starting at the top of the product taxonomy, and working down the $\mathcal{R}_{SC}$ relation from most general to most specific. It was not useful to treat the store taxonomy this way, as it is very shallow. When a pair was given an overall negative label, we did not sample any subcategories and inferred a negative label for all descendents (a few inferred-negative categories would get sampled if they had another parent category with a non-negative label). For example, since *Auto parts stores* do not sell *Grocery* and $\langle Dairy, Grocery \rangle \in \mathcal{R}_{SC}$, we did not ask ⟨*Auto parts stores, Dairy*⟩. *Electronics* are not sold at *Pharmacies* but *Batteries* are, and $\langle Batteries, Electronics \rangle \in \mathcal{R}_{SC}$, which would supress asking that pair. Fortunately, $\langle Batteries, HouseholdProducts \rangle \in \mathcal{R}_{SC}$, and *HouseholdProducts* are sometimes sold at *Pharmacies*, allowing us to ask $\langle Batteries, Pharmacies \rangle$.

This top-down taxonomic pruning eliminated any need for the web signals, and accounted for the sparsity at a very high level, since (by accident or ontology) the store and product taxonomies were well aligned: e.g. *Auto parts stores* sell *Auto parts* and do not sell *Groceries*. Higher level categories also made a lot more sense to raters when presented with a sub-category, e.g. *Sports & Outdoor Electronics* with *Fitness Trackers*, and since our rater pool did not vary much, they became familiar with the taxonomic distinctions as they progressed down the taxonomy over time.

Ambiguity

In the pilot experiments we observed disagreement in the results that did not support our crowd hypothesis, but were caused by ambiguity mainly in the product categories:

- product is a material, substance (e.g. plastic, starch, arugula) or some product aspect (e.g. color, size)
- product is a brand (e.g. Avian, Kleenex) or contains a brand name (e.g. Nike Sneakers, Todd's boots)
- store or product is too specific (e.g. duck sauce, goat meat, vanilla orchids, banner store)
- store or product is too generic (e.g. gift, organic food)
- product is regional (e.g. Harissa, Jajangmyeon)
- product is seasonal (e.g. christmas trees, flip-flops)
- product is polysemous in a way that is resolved by the store type, e.g. "fish" in a grocery store vs. a pet store

In MATRIX and SINGLETON, for example, raters seem more willing and able to answer the question, "Is milk sold here?" compared to "Is dairy sold here?" In the latter case, there is uncertainty over what minimum set of dairy items (milk, cheese, butter, yogurt, etc.) would be needed for "sells dairy" to be true, yet the equally rich sub-categories of milk (whole milk, skim milk, organic milk, etc.) did not cause the same uncertainty. When presented with the categories in a top-down fashion, raters first dealt with their uncertainty about "dairy" and applied it to the subcategories as well. For many store types, raters were willing to give definite answers about the other sub-types in subsequent tasks.

We specifically addressed the material, aspect and brand problems by removing them from the product set, their treatment is the subject of future work. We instructed the raters to treat seasonal products as "year round", after confirming that users are less likely to search for such products out of season. We updated the task design to allow raters to explore the two taxonomies, but we found that grouping store categories by taxonomic (sibling and parent) relations in PRODCAT obviated this exploration.

Regional products produced disagreement esp. across countries, where for the final tasks we sourced raters in five countries. Often this showed up merely as "I Don't Know" answers which were not used in predicting $\mathcal{R}_I$, but do show up in IRR. More interesting cases included when a product had a slightly different meaning, or was sold in different types of stores, in different regions. For example, "syrup" in France is sold in drug stores, and raters in other countries did not agree. This is because in France "syrup" is cough syrup, and this association did not exist elsewhere that we tested. We save further analysis of regional differences for a future paper, and absorbed these disagreements as inaccuracies in the experiments below.

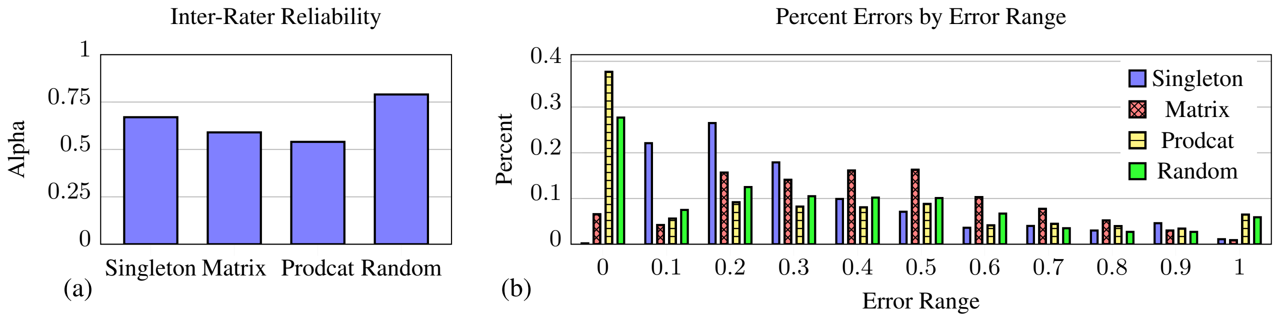
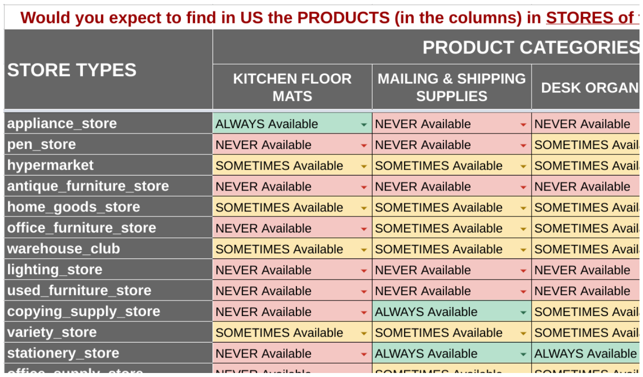


Figure 4: (a) Krippendorff-α and (b) normalized-MAE for the various task designs

PRODCAT Data Collection Task

The final design of the PRODCAT task presented a matrix of $\langle c_s, c_p \rangle$ pairs to raters in five countries, five raters per country, and consisted of several elements:

- a list of store categories, $c_s \in \mathcal{C}_S$
- a list of product categories, $c_p \in \mathcal{C}_P$
- c_s, c_p pairs presented in an $n \times 4$ matrix, where each c_s is a row and each c_p is a column; n ranged from 40-200 depending on our ability to find suitable products
- the matrix was prefaced with: "Would you expect to find in *country* the products (in the columns) in stores of the types (in the rows)?"
- each cell in the matrix connected one pair with four possible answers: "Always available", "Sometimes available", "Never available", and "I Don't Know"
- the row and column headers c_p and c_s included links to an image, a short description, and the position in the respective taxonomy
- raters were encouraged to explore the taxonomies in order to better understand categories
- The column product types were chosen such that three were taxonomy-related (sibling or more-specific child) and one was not, e.g. "aspirin", "notebooks", "paper supplies", "lined paper".



Would you expect to find in US the PRODUCTS (in the columns) in STORES of

STORE TYPES	PRODUCT CATEGORIES		
	KITCHEN FLOOR MATS	MAILING & SHIPPING SUPPLIES	DESK ORGAN
appliance_store	ALWAYS Available	NEVER Available	NEVER Available
pen_store	NEVER Available	NEVER Available	SOMETIMES Avail
hypermarket	SOMETIMES Available	SOMETIMES Available	SOMETIMES Avail
antique_furniture_store	NEVER Available	NEVER Available	NEVER Available
home_goods_store	SOMETIMES Available	SOMETIMES Available	SOMETIMES Avail
office_furniture_store	NEVER Available	SOMETIMES Available	SOMETIMES Avail
warehouse_club	SOMETIMES Available	SOMETIMES Available	SOMETIMES Avail
lighting_store	NEVER Available	NEVER Available	NEVER Available
used_furniture_store	NEVER Available	NEVER Available	NEVER Available
copying_supply_store	NEVER Available	ALWAYS Available	SOMETIMES Avail
variety_store	SOMETIMES Available	SOMETIMES Available	SOMETIMES Avail
stationery_store	NEVER Available	ALWAYS Available	ALWAYS Available
office_supply_store	NEVER Available	SOMETIMES Available	SOMETIMES Avail

Figure 5: Partial view of the PRODCAT data collection template with example answers from one rater

The final matrix PRODCAT crowd template is shown in Fig. 5 with an example of answers provided by one rater. Based on rater feedback and metrics shown in Sec. Reliability of Ratings and Sec. Error of Ratings, this presentation helped resolve many forms of polysemy mentioned in Sec. Ambiguity.

Reliability of Ratings

Tab. 2 shows a small sample of the CS task results for $\mathcal{R}_C$ pairs; we have intentionally downsampled the '5-never' pairs to show a mixture of different vote ratios.

Since the ratings of "Always", "Sometimes" and "Never" are ordered, we used Krippendorff's α to measure the inter-

category	product	always	some	never
auto parts store	pita	0	0	5
bakery	Longline Vests	0	0	5
beauty supply store	aromatherapy	5	0	0
bicycle store	home furnishings	0	0	5
butcher shop	quicklime	0	0	5
chinaware store	watches	0	0	5
clothing store	Women's Shirts	5	0	0
clothing store	Petite Negligee	5	0	0
clothing store	Truck Tailgate Caps	0	0	5
clothing store	chameleon	0	0	5
clothing store	typewriter ribbon	0	0	5
coffee store	Instant Coffee	4	0	1
cosmetics store	Non-Dairy Milk	0	0	5
drugstore	tarragon	0	0	5
electronics store	Canister Vacuums	5	0	0
feed store	cybex	0	0	5
fresh food market	Work Dresses	0	0	5
fruits & vegetables	Turkey Sausage	0	1	4
furniture store	Canopy Beds	4	1	0
furniture store	Box Springs	4	0	1
grocery store	Smart Light Bulbs	0	0	5
grocery store	Frozen Clams	5	0	0
grocery store	soy nuts	4	1	0
home goods store	Storage Baskets	4	1	0

Table 2: Example CrowdSense Ratings on $\mathcal{R}_C$ pairs

rater reliability, with the usual distance function:

$$dist(x,y) = \begin{cases} 0, & \text{if } x = y \\ 0.5, & \text{if } x = \text{Sometimes} \vee y = \text{Sometimes} \\ 0.5, & \text{if } x = \text{Unknown} \vee y = \text{Unknown} \\ 1, & \text{otherwise} \end{cases}$$

Fig. 4(a) presents α scores for the various tasks. The RANDOM baseline scored highest since its purpose was to confirm our sparsity estimate – with 95% items receiving all 5 ratings as "Never", high agreement resulted. The SINGLETON task was guided by the web signals, leading to pairs that were more likely to be obviously "Always", though much more of a mixture than RANDOM. For MATRIX, we began to actually explore the space of $\mathcal{R}_C$ where disagreement indicates the distribution of $\mathcal{R}_I$, and this is carried further in the sampling used by PRODCAT. Hence, it makes sense that the later task designs produced more disagreement, because we targeted pairs that would have it. This makes IRR a less than suitable measure for the adequacy of the MATRIX and PRODCAT tasks.

Error of Ratings

Since IRR cannot reflect the quality of ratings where disagreement is the desired result, we measure the error of different $\mathcal{R}_C$ pairs in predicting the distribution of $\mathcal{R}_I$ pairs, by comparing ratings-based scores on $\mathcal{R}_C$ pairs against UGC scores on $\mathcal{R}_I$ pairs obtained from users[8] (see Sec. Answers from Users). Each class and instance level pair has a score:

$$w_{x,p} = \begin{cases} (\alpha_{x,p} - \nu_{x,p})/(\alpha_{x,p} + \nu_{x,p}) & \text{if } x \in \mathcal{C}_S \\ (y_{x,p} - n_{x,p})/(y_{x,p} + n_{x,p}) & \text{if } x \in \mathcal{I}_S \end{cases}$$

where $\alpha_{x,p}$ is the number of "always" answers for class-level pairs $\langle x, p \rangle$ and $\nu_{x,p}$ the number of "never" answers; and $y_{x,p}$ is the number of "yes" answers for store instance-level pairs $\langle x, p \rangle$ and $n_{x,p}$ the number of "no" answers.

Next let $\mathcal{I}_c = \{i : \langle i, c \rangle \in \mathcal{R}_T\}$ be the instances of category c under R_T. The mean absolute error of $\langle c, p \rangle$ is:

$$\textsc{mae}(\langle c, p \rangle \in \mathcal{R}_C) = \frac{\sum_{i \in \mathcal{I}_c} |w_{i,p} - w_{c,p}|}{|\mathcal{I}_c|}$$

The idea is that if the class-level scores ($w_{c,p}$) are an accurate prediction of the availability distribution at the instance level, then they should model user observations at individual stores ($w_{i,p}$), averaged over the size of the store category ($|\mathcal{I}_c|$). Fig. 4(b) shows the distribution of MAE scores per category pairs for each of the four data collection tasks. Despite PRODCAT being a harder task due to the sampled pairs, it performs much better than the other tasks, with nearly half of its categories scoring in the lowest error range, clearly supporting our crowd hypothesis: the disagreement on $\langle c_s, c_p \rangle$ pairs approximates the distribution of $\langle i_s, c_p \rangle$ when $\langle i_s, c_s \rangle \in \mathcal{R}_\mathcal{T}$, according to user observations.

Instance-Level Prediction Experiments

Data Sources

We compare and contrast several approaches for acquiring and predicting the relations in $\mathcal{R}_I$:

CrowdSense (CS): 25k pairs of class-level associations $\langle c_s, c_p \rangle \in \mathcal{R}_C$ and an associated score for each pair w_{c_s,c_p}, collected through PRODCAT (as described above). In our experiments, we treated the CS data as a static set, although in practice it could grow or change over time like UGC.

User Responses (UGC): As described in Sec. Answers from Users, we collected more than 100M instance-level pairs from volunteer users around the world over a two year period.[9] Most of the UGC pairs have a distribution of yes and no answers, and more sophisticated processing of the answers is possible, but for simplicity we use the majority vote as the label in the experiments below, where we break the data into sets representing the first $n \in [1, 24]$ months of collection, to illustrate the growth of the data over time.

Web baseline (WebIE): The baseline approach to supporting *local shopping* queries is the Web: using product names mentioned on store web pages as part of an inverted index that are matched to search queries for those products. As discussed above, this approach for local shopping is limited by the coverage of local (aka brick & mortar) stores on the web, which was under 30% (60% for the US) at the start of this project in 2017, and has not increased substantially in the years hence. WebIE is only able to obtain positive labels, leaving negatives to be inferred from the complement.

To serve this paper, we isolated the product mention signal from others used in modern web search (e.g. click-throughs, co-occurrence, etc.) to produce the WebIE dataset, collecting only the instance-level pairs in $\mathcal{R}_I$. We used a named entity recognizer to extract instance-level pairs ($\mathcal{R}_I : \mathcal{I}_S \times \mathcal{C}_P$) for stores with a web site that mention products on any of the site's pages, and used the extraction confidence probability. While other web sources (user reviews, coupons, photos, etc.) and more advanced product extraction techniques (see (Wang et al. 2020)) might improve the recall for web data, for most stores, this information simply is not available, and this baseline is a very good representation of that. We treated the Web dataset as a single unchanging set; for our experiments, the change over time was not significant enough to measure.

WALS(UGC): Since predictions of the instance-level pairs form a matrix, $\mathbf{R}_{s,p}$, an obvious approach is to use matrix factorization on the matrix formed by one of the above methods. We used an off-the-shelf WALS(Koren, Bell, and Volinsky 2009) implementation on the UGC scores discussed below. Since WALS does not use "features", but rather a matrix of real values, we did not include other inputs to WALS in Fig. 6.

Evaluation

Ultimately our goal is to enable product queries to return nearby stores on maps as well as (web) search results; however, direct application impact metrics from our system, which launched in mid-2020, are proprietary. Here we focus on metrics for the part of the system dealing with knowledge acquisition as one of knowledge-based completion(McNamee and Dang 2009; Welty et al. 2012).

[8] We used user-generated content (UGC) for evaluation due to the absence of a large, uniformly-sampled ground-truth dataset.

[9] Collection continues, this two-year window was used for this paper

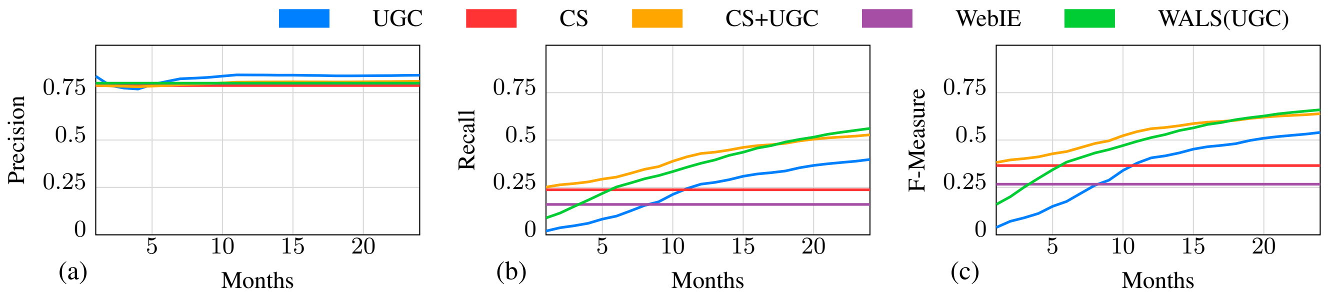


Figure 6: Precision, Recall, and F-measure for different ways of predicting $\mathcal{R}_I$.

We collected 40k gold standard $\langle i_s, c_p \rangle$ pairs by calling each store i_s asking them if they sold c_p. The stores were selected from among more than 50 countries with the top-5 countries being US (20%), JP (5%), IN (5%), GB (5%), BR (4%); stores within each country were sampled uniformly to provide a microcosm of representative demographics. We used these pairs as a test set in the experiments below. As with the CS and UGC approaches, the sparsity of $\mathcal{R}_I$ makes uniformly sampling pairs wasteful. To achieve better class balance, the WebIE baseline data was used to guide the collection towards pairs that had an increased chance of being true; for example, if a store's webpage mentioned a product we would try to call stores of the same type and ask about that product. We targeted a positive/negative class balance of 50%, and targeted a stratification of the sampling that preserved the 30/70 balance of stores with and without websites. When evaluating against the gold standard, any instance-level pairs that are present in the gold set but missing in the evaluated data are counted as negatives towards recall. Tab. 3 shows a small sample of these.

Results

WebIE Since the values on the WebIE data for each $\langle i_s, c_p \rangle \in \mathcal{R}_I$ are fractional in $[0, 1]$, we determined the lowest threshold with at least 0.80 precision and computed recall based on that, resulting in a recall of 0.136 at 0.80 precision. This recall reflects the fraction of the stores with web pages, the fraction of products mentioned on those pages, and the recall of the product named entity recognition. We did not independently measure these other factors, as Web performance was merely a baseline. We also used WALS to infer values for other instance-level pairs, improving recall to 0.151 at 0.80 precision (not shown).

CS The primary hypothesis of this paper is the acquisition of class-level associations in $\mathcal{R}_C$ from the crowd is an effective way of rapidly jump-starting instance-level associations in $\mathcal{R}_I$. As described in Crowd Sense, we acquired 25k class-level pairs from a paid crowd, each with a score $w_{c,p}$ (see Error of Ratings), and chose the following simple procedure to infer the instance level pairs:

$$w_{c,p} > 0 \wedge c \in \mathcal{C}_S \implies \langle c, p \rangle \in \mathcal{R}_C$$
$$\langle c, p \rangle \in \mathcal{R}_C \wedge \langle i, c \rangle \in \mathcal{R}_T \implies \langle i, p \rangle \in \mathcal{R}_I$$

We then measured the effectiveness of the CS by comparison of the inferred edges in $\mathcal{R}_I$ to the Gold set, achieving a recall of 0.238 with a precision of 0.788. While this shows a distinct improvement over WebIE, of more interest is the combination, which improves recall to 0.351 – near perfect complementarity – while slightly losing precision at 0.782 (for simplicity we do not show this in Fig. 6). The combination uses the WebIE or CS signal if the other is not present, and the CS signal if they are both present, since the CS data includes negatives and WebIE does not. (WALS inference was ineffective here; see below.)

UGC The UGC dataset grows over time, while we treat the Web and CS data as constant (see above). We expect that, given enough time, UGC will overtake CS and WebIE in recall, so an important question is how much time the CS data is worth compared to UGC, and whether it continues to show value. In Fig. 6, the blue line shows the precision, recall, and F1 score of the UGC data using the majority vote as the label, and the red line shows the CS performance, which, as noted above, doesn't change. At around 11 months, the UGC line crosses the CS line, indicating that CS is worth about 11 months of UGC collection.

WALS(UGC) We populated the matrix $\mathbf{R}_{s,p}$ from UGC $w_{s,p}$ scores, factorized $\mathbf{R}$ using WALS, and measured the resulting dot-products against the Gold Standard dataset choosing the 0.8 prec. threshold, shown in Fig. 6 in green. Note that some of the $\langle s, p \rangle$ pairs in the Gold set were in the training set, however the *labels* used in the training matrix may be different than Gold, making it a fair comparison. As in the previous experiments we broke the dataset into sets representing the first $n \in [1, 24]$ months of collected user

store	**category**	**loc**	**product**	**available**
7-Eleven	convenience store	US	distilled water	FALSE
ALDI	grocery store	US	fruitcake	TRUE
AURORA MKT	store	US	Men's Gloves	FALSE
Adams Pharmacy	pharmacy	US	kool aid	TRUE
Ag construcciones	building materials	PY	Blinds	TRUE
Alanyurt Gıda	general store	TR	Razor Blades	TRUE
Amorino	ice cream shop	FR	meat	FALSE
Barnes & Noble	book store	US	blankets	FALSE
Barstow Buick	car dealer	US	crown victoria	TRUE
Barstow Buick	car dealer	US	gears	TRUE
Bazar	bazar	BR	mary kay	FALSE

Table 3: Example gold standard $\mathcal{R}_I$ pairs

responses. WALS clearly improves over UGC.

CS+UGC While 11 months is the intersection point of the metric values for CS and UGC independently, the CS data is supposed to complement as well as jump-start the knowledge acquisition. We tested the role of CS over time using a simple "CS as default" combination, shown in Fig. 6 as CS+UGC, in which the UGC label is used if present, and the CS label is used if not. This line tracks the improvement in recall over time from UGC collection, while jump starting at the recall of CS. This is a clear demonstration of our core research hypothesis.

Of particular interest is the comparison of WALS(UGC) with CS+UGC. The former does eventually surpass the latter after roughly 18m, but the CS+UGC combination is a strong contender from an extremely simple method. This is again clear evidence of our core hypothesis. Other ways of filling the initial training matrix $\mathbf{R}_{s,p}$ by combining CS, UGC, and WebIE signals in various ways were tried but not included as they do not outperform WALS(UGC). Of note is that the CS signal does not work well with WALS, since it effectively does what WALS itself should do with enough data - filling in giant portions of the matrix with default values. Other machine learning approaches are certainly possible, indeed the launched *local shopping* system uses a deep neural network with many more features that are beyond the scope of this paper, and measured at the scale of the web. The three signals reported here are very signifant features of that system, and the full system improves significantly over search alone.

Related Work

The core of this work is overcoming a *knowledge acquisition* bottleneck in acquiring data reflecting the availability of products at millions of brick&mortar stores worldwide. The approach of harnessing class-level knowledge to the infer instance-level knowledge is based on a long standing idea in knowledge engineering (Minsky 1974). Other methods in the formal *knowledge represention* (KR) field have never scaled to the level necessary for our problem, nor have they considered the problem of how to acquire distributions instead of discrete facts.

Information Extraction (IE) methods perform knowledge acquisition of real-world entities from web text, and are discussed in (Zang et al. 2013). (Martínez-Rodríguez, Hogan, and López-Arévalo 2020) presents a survey of IE techniques for populating semantic structures, e.g. entity extraction and linking. In the context of shopping, research has mainly focused on product information extraction, e.g. crawling the Web for offers to maintain product catalogs (Nguyen et al. 2011; Qiu et al. 2015a) with product specifications and attributes (Qiu et al. 2015b; Kannan et al. 2011; Zheng et al. 2018; Wang et al. 2020), and IE methods for building product knowledge graphs (Dong and al. 2020; Xu et al. 2020). Our paper defines a method for linking these already defined entities similar to (Dong and al. 2020), incorporating product and store taxonomy knowledge.

Knowledge Base Completion (KBC)is the problem of inferring missing entities and/or relations in an existing knowledge graph based on existing ones, such as via link prediction (Bordes et al. 2013) or from a combination of sources (Riedel et al. 2013). Our product × store category matrix (Fig. 5) is inspired by the item-based collaborative filtering matrix introduced in recommender systems (Sarwar et al. 2001; Ekstrand, Riedl, and Konstan 2011), and we leverage a well-known collaborative filtering approach (Koren, Bell, and Volinsky 2009) for KBC to demonstrate the additional power of inference on our knowledge graph.

We use a knowledge graph as the basic representation and, like most well known KGs, employ no general-purpose reasoning; hence, any inference we do must be defeasible. The most relevant KR area would be reasoning with defaults e.g. (Lang 2000), as our CS+UGC baseline mechanism for combining $\langle c_s, c_p\rangle$ with $\langle i_s, c_p\rangle$ pairs treats the first as a default and the second as an override. Beyond this simple combination strategy, which was first proposed by Quillian in 1967 (Quillian 1967), more sophisticated combinations of CS+UGC with other forms of evidence are done using optimizations from machine learning. The full *local shopping system* uses many signals, of which we've described only three, that are combined using a deep neural network that optimizes the prediction of observed labels for many billions of $\langle i_s, c_p\rangle$ pairs. While we exploit the taxonomies in $\mathcal{C}_\mathcal{S}$ and especially $\mathcal{C}_\mathcal{P}$ to optimize the selection of class-level pairs to acquire from workers (Lees et al. 2020), taxonomy-based reasoning was only used for negative associations.

IE and KBC techniques have advanced the state-of-the-art in capturing human knowledge in machine-readable form, but there is still the need for human curation and *crowdsourcing*. Important milestones for crowdsourcing knowledge acquisition at scale are Wikidata (Vrandečić and Krötzsch 2014) and Freebase (Bollacker et al. 2008), where the crowd defines or curates real world entities and some relationships between them, typically driven by Wikipedia. With respect to KBC, (Revenko et al. 2018) proposes a method for crowdsourcing categorical common sense knowlegde from nonexperts for adding new relationships between nodes in the graph and ensuring consistencey with existing relations. However in all these sources, the sparsity of graph edges expressing relations between the class-level nodes is high (Taylor 2017). Our work focuses directly on that problem by acquiring both class-level and instance level graph edges, and scaling the latter from the former.

The crowdsourcing approach we propose in this paper is grounded in the theoretical framework of (Aroyo and Welty 2013, 2014), which breaks the constraints of typical methodologies for collecting ground truth, showing disagreement is a necessary characteristic of annotated data; when interpreted correctly it can make evaluation of machine learning models more attuned to real-world data (Dumitrache 2019).

The immense body of research on common sense and crowdsourcing has directly influenced our work. The UGC and Crowd Sense tasks drew on our knowledge of Games-with-a-purpose such as Verbosity(von Ahn, Kedia, and Blum 2006) for collecting common sense facts, Common Consensus(Lieberman, Smith, and Teeters 2007) for gathering common sense goals, GECKA(Cambria et al. 2016) for common sense knowledge acquisition, Concept Game (Herdagdelen and Baroni 2010) for verifying common sense knowledge assertions, the FACTory Game (Lenat and Guha 1989) for facts verification and many others. (Rodosthenous and Michael 2019) refer to common sense as ”knowledge about the world” and propose a hybrid (machine and human tasks) workflow to gather general common sense knowledge rules.

Active learning investigates efficiency for acquisition and learning when acquiring training data for ML models. In essence, the early stages of KG acquisition strongly represent the exploration side of the *exploration vs. exploitation* tradeoff (Bondu, Lemaire, and Boullé 2010). ML models during exploration do not have enough knowledge of the space to be able to offer reliable judgements as to which items (in this case, $\langle i_s, c_p \rangle$ pairs) to acquire labels for. As noted in the Data Sources section, class-level pairs can serve as a guide for recognizing obvious $\langle i_s, c_p \rangle$ pairs that likely do not need labels, and conversely, high-disagreement pairs are very likely to have instances that do. Thus the $\langle c_s, c_p \rangle$ pairs can serve to stratify the $\langle i_s, c_p \rangle$ space, and make the job of active learning easier by narrowing down their targets.

Perhaps the most similar crowdsourcing work to ours studies the problem of approximating aggregation queries (Trushkowsky et al. 2013), such as “How many restaurants in San Francisco serve scallops?” While this approach works well for estimating counts, clearly it does not scale for KBC.

Conclusions

The Shopping Sense project and the CrowdSense approach were integral parts of a successful worldwide launch of local shopping results overlaid on Google Maps, as shown in Fig. 1. More recently, we launched local *dining* results that responds to searches for dish names with restaurants that serve that dish.

Due to the complexity and scope of the deployed project, we focused on the real-world knowledge acquisition aspect of the work, and presented a few simplified experiments that demonstrate how the acquired class-level knowledge can be used for KBC at the instance level. These experiments may seem over-simplified, but they accurately capture the impact of the three-tiered crowdsourcing approach on the deployed product, in particular the rapid jump-start of the place-product edges in the knowledge graph.

To achieve these results, we augmented an existing knowledge graph of most stores on earth, their categories, and a product taxonomy, by adding store to product edges. We combined web-based information extraction (WebIE) and direct user observations (UGC) with a novel collection of class-level $\langle store, product \rangle$ pairs from the crowd (CS) which were inferred to the instance-level based on class membership. In two weeks of data collection we achieved a recall of 0.24 at 0.80 precision against gold standard instance-level labels, combining with WebIE to achieve 0.35 recall, which was the recall of UGC after 20 months, and of a WALS model with UGC input in 18 months. We conclude that the Crowd Sense approach uses human common sense knowledge to *rapidly jump start* the kind of generalization that ML systems are good at with a lot of data. This has implications for practical ML and Human Computation.

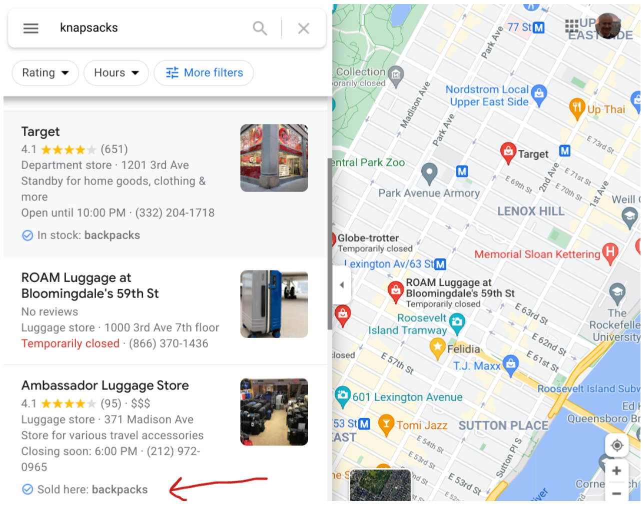
Our class-level crowdsourcing results show that the disagreement in categorical knowledge collected from the crowd can indicate the distribution of that knowledge at the instance level, rather than assuming the class-level associations are universally true: in other words, if 80% of raters say “Grocery stores sell oat milk”, then $\sim 80\%$ of grocery stores sell oat milk. These results held also for dishes at restaurants.

The taxonomy of products (and dishes) was used to guide the sampling of class-level pairs in a way that helped us address the sparsity of the $\mathcal{C}_S \times \mathcal{C}_P$ space, and only the *negative* class-level attributes were accurate when inferred to more specific categories, as opposed to the more traditional view that positive attributes are “inherited.”

We found the categorical pairs which were rapidly acquired were extremely useful in guiding the collection of instance-level labels, since we did not have to ask users about obviously available or unavailable products – this has implications for active learning, and held also for dining.

Crowd Sense generalizes to other bipartite relations between places and types of entities that are organized in a taxonomy, such as dishes at restaurants, services at professional offices, etc., as well as a wide range of other bipartite graph problems where common sense or categorical knowledge prevails as defaults, such as ingredients for dishes, linnean taxonomies of living creatures, etc.

To see CrowdSense at work, type the name of a product or dish into Google Maps (or Google Search). Results that say “Sold here: *product*” come from the data we published, as opposed to “In stock” (merchant feeds) and “Webpage says:”. Anyone with a Google account can participate in UGC (user generated content) acquisition. Users with location tracking turned on (so that maps knows what places the user has visited[10]) can navigate to the ‘contribute’ tab that allows them to rate and leave reviews, as well as review facts and answer the yes/no questions regarding locations they have visited.



Target
4.1 ★★★★☆ (651)
Department store · 1201 3rd Ave
Standby for home goods, clothing & more
Open until 10:00 PM · (332) 204-1718
In stock: **backpacks**

ROAM Luggage at Bloomingdale's 59th St
No reviews
Luggage store · 1000 3rd Ave 7th floor
Temporarily closed · (866) 370-1436

Ambassador Luggage Store
4.1 ★★★★☆ (95) · $$$
Luggage store · 371 Madison Ave
Store for various travel accessories
Closing soon: 6:00 PM · (212) 972-0965
Sold here: **backpacks**

Figure 7: CrowdSense search results in NYC for knapsacks

[10] See https://maps.google.com/localguides/howto

References

Aroyo, L.; and Welty, C. 2013. Crowd Truth: Harnessing disagreement in crowdsourcing a relation extraction gold standard. In *Proceedings of the 5th Annual ACM Web Science Conference*, WebSci '13.

Aroyo, L.; and Welty, C. 2014. The Three Sides of CrowdTruth. *Human Computation* 1(1): 31–44.

Aroyo, L.; and Welty, C. 2015. Truth is a lie: Crowd Truth and the seven myths of human annotation. *AI Magazine* 36(1): 15–24.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proc. SIGMOD International Conference on Management of Data*, 1247–1250. Association for Computing Machinery.

Bondu, A.; Lemaire, V.; and Boullé, M. 2010. Exploration vs. exploitation in active learning : A Bayesian approach. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*.

Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; and Yakhnenko, O. 2013. Translating Embeddings for Modeling Multi-relational Data. In Burges, C. J. C.; Bottou, L.; Ghahramani, Z.; and Weinberger, K. Q., eds., *NIPS 2013*, 2787–2795.

Cambria, E.; Nguyen, T. V.; Cheng, B.; Kwok, K.; and Sepulveda, J. 2016. GECKA3D: A 3D Game Engine for Commonsense Knowledge Acquisition. In *CoRR-2016*.

Dong, X. L.; and al. 2020. AutoKnow: Self-Driving Knowledge Collection for Products of Thousands of Types. In *KDD '20: Conference on Knowledge Discovery and Data Mining*, 2724–2734. ACM.

Dumitrache, A. 2019. *Truth in Disagreement: Crowdsourcing Labeled Data for Natural Language Processing*. Ph.D. thesis, VU Amsterdam.

Ekstrand, M. D.; Riedl, J. T.; and Konstan, J. A. 2011. *Collaborative filtering recommender systems*. Now Publishers Inc.

Herdagdelen, A.; and Baroni, M. 2010. The Concept Game: Better Commonsense Knowledge Extraction by Combining Text Mining and a Game with a Purpose. In *AAAI Fall Symposium: Commonsense Knowledge*.

Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Yu, P. S. 2020. A Survey on Knowledge Graphs: Representation, Acquisition and Applications. *CoRR* abs/2002.00388. URL https://arxiv.org/abs/2002.00388.

Kannan, A.; Givoni, I. E.; Agrawal, R.; and Fuxman, A. 2011. Matching Unstructured Product Offers to Structured Product Specifications. In *KDD-2011*, 404–412.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42(8): 30–37.

Lagos, N.; Ait-Mokhtar, S.; and Calapodescu, I. 2020. Point-Of-Interest Semantic Tag Completion in a Global Crowdsourced Search-and-Discovery Database. In Giacomo, G. D.; Catalá, A.; Dilkina, B.; Milano, M.; Barro, S.; Bugarín, A.; and Lang, J., eds., *ECAI 2020*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, 2993–3000. IOS Press.

Lang, J. 2000. *Possibilistic Logic: Complexity and Algorithms*, 179–220. Springer Netherlands.

Lees, A. W.; Welty, C.; Korycki, J.; Carthy, S. M.; and Zhao, S. 2020. Embedding Semantic Taxonomies. In *CoLing 2020*.

Lenat, D. B.; and Guha, R. V. 1989. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc.

Lieberman, H.; Smith, D.; and Teeters, A. 2007. Common Consensus: a web-based game for collecting commonsense goals. In *IUI-2007*.

Martínez-Rodríguez, J. L.; Hogan, A.; and López-Arévalo, I. 2020. Information extraction meets the Semantic Web: A survey. *Semantic Web* 11: 255–335.

McNamee, P.; and Dang, H. T. 2009. Overview of the TAC 2009 knowledge base population track. In *Text Analysis Conference (TAC-2009)*.

Minsky, M. 1974. A framework for representing knowledge. Technical report, MIT.

Nguyen, H.; Fuxman, A.; Paparizos, S.; Freire, J.; and Agrawal, R. 2011. Synthesizing Products for Online Catalogs. *Proc. VLDB Endow.* 4(7): 409–418.

Qiu, D.; Barbosa, L.; Dong, L. X.; Shen, Y.; and Srivastava, D. 2015a. DEXTER: Large-Scale Discovery and Extraction of Product Specifications on the Web. *PVLDB* 2194–2205.

Qiu, D.; Barbosa, L.; Dong, X. L.; Shen, Y.; and Srivastava, D. 2015b. DEXTER: Large-Scale Discovery and Extraction of Product Specifications on the Web. *Proc. VLDB Endow.* 8(13): 2194–2205.

Quillian, M. R. 1967. Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Science* 12(5).

Revenko, A.; Sabou, M.; Ahmeti, A.; and Schauer, M. 2018. Crowd-Sourced Knowledge Graph Extension: A Belief Revision Based Approach. In Bozzon, A.; and Venanzi, M., eds., *Proceedings of the HCOMP 2018 Works in Progress and Demonstration Papers Track*, volume 2173 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Riedel, S.; Yao, L.; Marlin, B. M.; and McCallum, A. 2013. Relation Extraction with Matrix Factorization and Universal Schemas. In *Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*.

Rodosthenous, C.; and Michael, L. 2019. A Platform for Commonsense Knowledge Acquisition Using Crowdsourcing. In *Proc. of the enetCollect WG3 & WG5 Meeting 2018*, volume Vol-2390, 25–30.

Sarwar, B.; Karypis, G.; Konstan, J.; and Riedl, J. 2001. Item-Based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, 285–295. Association for Computing Machinery.

Surowiecki, J. 2005. *The Wisdom of Crowds*. Anchor. ISBN 0385721706.

Taylor, J. 2017. ISWC 2017 Keynote: Applied semantics: beyond the catalog. https://iswc2017.ai.wu.ac.at/program/keynotes/keynote-taylor/.

Trushkowsky, B.; Kraska, T.; Franklin, M. J.; and Sarkar, P. 2013. Crowdsourced enumeration queries. In *29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013*, 673–684. IEEE Computer Society.

von Ahn, L.; Kedia, M.; and Blum, M. 2006. Verbosity: a game for collecting common-sense facts. In *Proc. Conference on Human Factors in Computing Systems, CHI 2006*, 75–78. ACM.

Vrandečić, D.; and Krötzsch, M. 2014. Wikidata: A Free Collaborative Knowledge Base. *Communications of the ACM* 57: 78–85.

Wang, Q.; Yang, L.; Kanagal, B.; Sanghai, S.; Sivakumar, D.; Shu, B.; Yu, Z.; and Elsas, J. 2020. Learning to Extract Attribute Value from Product via Question Answering: A Multi-task Approach. In *KDD-20*.

Welty, C.; Barker, K.; Aroyo, L.; and Arora, S. 2012. Query driven hypothesis generation for answering queries over nlp graphs. In *The Semantic Web–ISWC 2012*, 228–242. Springer.

Xu, D.; Ruan, C.; Korpeoglu, E.; Kumar, S.; and Achan, K. 2020. Product Knowledge Graph Embedding for E-Commerce. In *Proc. of Conference on Web Search and Data Mining*, WSDM '20, 672–680.

Zang, L.; Cao, C.; Cao, Y.; Wu, Y.; and Cao, C. 2013. A Survey of Commonsense Knowledge Acquisition. *J. Comput. Sci. Technol.* 28(4): 689–719.

Zheng, G.; Mukherjee, S.; Dong, X. L.; and Li, F. 2018. OpenTag: Open Attribute Value Extraction from Product Profiles. In *Proc. KDD '18*, 1049–1058. Association for Computing Machinery.