Matching Physical Sites with Web Sites for Semantic Localization

Rufeng Meng[*+] Sheng Shen[*] Romit Roy Choudhury[*] Srihari Nelakuditi[+]
[*]University of Illinois (UIUC) [+]University of South Carolina (USC)

ABSTRACT

Locations are often expressed in physical coordinates such as an $[X, Y]$ tuple in some coordinate system. Unfortunately, a vast majority of location-based applications desire the semantic translation of coordinates, i.e., store-names like Starbucks, Macy's, Panera. Past work has mostly focused on achieving localization accuracy, while assuming that the translation of physical to semantic coordinates will be done manually. In this paper, we explore an opportunity for automatic semantic localization – the presence of a website corresponding to each physical store. We propose to correlate the information seen in a physical store with that found in websites of the stores around that location, to recognize that store. Specifically, we assume a repository of crowdsourced WiFi-tagged pictures from different stores. By correlating words inside the pictures, against words extracted from store websites, our proposed system can automatically label clusters of pictures, and the corresponding WiFi APs, with the store name. Later, when a user enters a store, her smartphone can scan the WiFi APs and consult a lookup table to recognize the store she is in. Our preliminary experiments with 18 stores in a shopping mall show that, our prototype system could correctly match the text from the physical stores with the text extracted from the corresponding web sites and hence label WiFi APs with store names with an accuracy upwards of 90%, which encourages us to pursue this study further. Moreover, we believe the core idea of correlating physical and web sites has broader applications beyond semantic localization, leading to better product placement and shopping experience, yielding benefits for both store owners and shoppers.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software; I.7.0 [**Document and Text Processing**]: General

Keywords

Semantic Localization; Textual Correlation; Crowdsourcing

WPA'15, May 22, 2015, Florence, Italy.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3498-3/15/05 ...$15.00.
http://dx.doi.org/10.1145/2753497.2753501.

1. INTRODUCTION

Indoor localization has been researched extensively in the last ten years. Even though a solution is not in the mainstream, localizing a user with $< 5m$ accuracy is possible today in the research community. While there is still research to be done in improving robustness, there is growing agreement that location accuracy and robustness are not sufficient to roll out localization in the wild. Indoor maps and semantic understanding of a place (e.g., Starbucks, Macy's, Panera) are equally crucial pieces of the bigger localization puzzle, but only recently researchers began to focus on these aspects.

This paper focuses on the problem of semantic localization of shoppers, i.e., making a smartphone aware of the store its user is in. Towards that end, we leverage the presence of a website corresponding to each physical store. Our approach is to correlate the information seen in a physical store with that found in websites of the stores around that location, to recognize that store. Specifically, the problem we are trying to address can be stated as follows: *Develop a system that detects the name of the store a user is in*, using a repository of crowd-sourced pictures from different stores, each picture tagged with WiFi APs. Put differently, can we automatically label WiFi APs with semantic store–names, using only pictures from different stores? If we can, then, when a new user enters a store, her smartphone can sense the WiFi APs and perform a simple lookup to identify the store name. Figure 1 illustrates the problem pictorially.

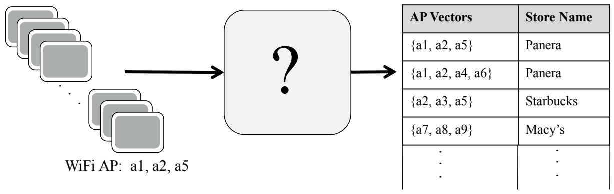

Figure 1: Problem Definition: A system is needed that receives WiFiAP-tagged pictures as input and generates a WiFiAP–StoreName lookup table as output.

One could incentivize crowd-sourced users to walk into different stores, record WiFi APs inside them, label the stores, and thus produce a WiFiAP–StoreName table manually. Our aim is to automatically label WiFi AP vectors with store names with the aid of in-store pictures gathered from users, without manual intervention, through a camera app or a photo storage service in the cloud. As an alternative, we investigated the possibility of inferring the store names by mining WiFi SSIDs [1]. Our results from the stores in the Champaign-Urbana county indicate that less than 30% stores indeed have meaningful AP names (e.g., Expresso Royale, Panera, Bestbuy-guest), from

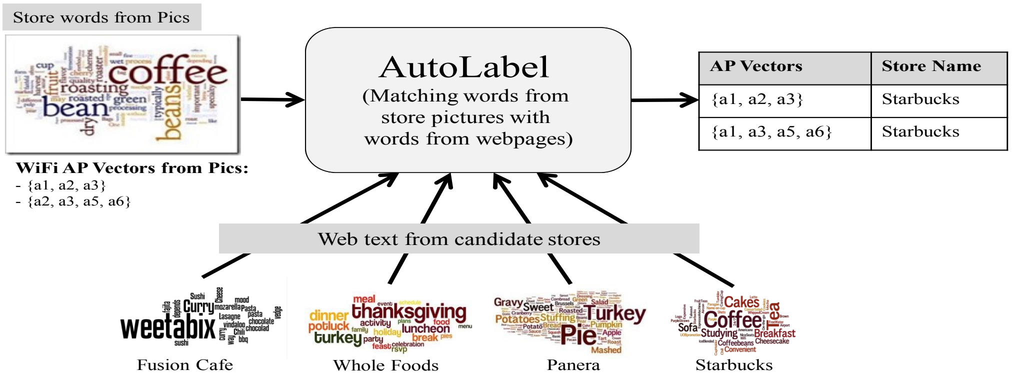

Figure 2: The core intuition underlying our system, AutoLabel.

which the store names could be guessed easily. An overwhelming majority are unrelated to the stores. Besides, in a large shopping mall, public WiFi service is provided by the shopping mall authority, therefore most of the stores there do not have their own APs. As a result, the reality of SSID naming and WiFi deployment has rendered the idea of mining store names from WiFi SSIDs inadequate.

Given a collection of in-store pictures, we immediately considered the possibility of finding logos and names of the stores in them. If a picture from Starbucks had the word "Starbucks" written in it, or the distinct green logo of the mermaid, it would be immediately possible to label the corresponding WiFi APs. Unfortunately, the approach presented hurdles. For instance, pictures from a "Payless" shoe store often had numerous logos of Nike and Adidas, similar to an actual Nike or Adidas store. Furthermore, when looking into the actual data, we realized that with many stores, not a single picture from the store had the logo or the store-name in it. Hence, an appropriate solution simply cannot rely on the store's name or logo being present in pictures.

This paper proposes to leverage the presence of an online version of each offline store and their similar content for recognizing a store given the in-store pictures. Our core intuition emerges from a hypothesis that different words visible in pictures from a store X, are also likely to occur in the website of the same store X. Put differently, let $H_{off}(Starbucks)$ denote the histogram of words visible from the pictures of Starbucks, and $H_{web}(i)$ denote a word histogram from store i's website. We hypothesize that when $H_{off}(Starbucks)$ is matched against $H_{web}(i)$, the strongest match will correspond to $i = Starbucks$. Figure 2 illustrates the idea. Of course, the matching need not be performed for all stores in the world – given that a picture's crude location is also known, only stores around that location can be candidates. Within this candidate set, the probability of a correct matching could be high.

Of course, the above may seem reasonable only if the crowdsourced pictures are already clustered per-store – in that case, words from each cluster can be labeled using the matching technique in Figure 2. Unfortunately, the picture repository may be flat, i.e., we may not have information on which pictures are from the same store. However, we believe that even without this information, the end goal can be achieved through a technique called *simultaneous clustering and labeling*.

We have developed a prototype of the proposed system, called AutoLabel, and evaluated it in a mall with 18 stores. Our experiments show labeling accuracy around 94%, even with as few as 10 valid pictures (that contain readable text) from a store, encouraging us to explore this approach further.

Generalization: The core connection between physical and web sites could potentially be applicable to broader problems beyond localization. Perhaps, user behavior on the website, say the ordering of visited items, could inform how products are laid out in the actual store. Conversely, interactions of shoppers with service agents in a store, difficult to capture from an e-commerce website, might prove valuable in improving the online experience. In a sense, this paper is merely a first step in exploiting the presence of two "avatars" of the same store. Next, we present the design of AutoLabel.

2. SYSTEM DESIGN

The key components of AutoLabel can be sketched as shown in Figure 3. The crowdsourcing app helps us collect images with the corresponding AP information. Using this collection of images, the AutoLabel server constructs WiFiAP−StoreName lookup table, which is used by an end-user app to semantically localize users. We elaborate on each of these components below.

2.1 In-store Data Extraction

AutoLabel aims to recognize a store based on the text from in-store pictures. Words from pictures can be extracted using optical character recognition (OCR) tools available publicly. Since our focus in this work is on studying how well the text in a store correlates with the text on its website, not on improving or evaluating OCR performance, we assume ideal conditions for extracting the in-store text. Among the OCR tools we tried, we found that Google Goggles [2] achieves the best performance in reading the text in pictures. At this moment, Google Goggles is provided as an app, Google hasn't opened its APIs. Therefore, in the current phase, we run the app on an Android phone, rotate the phone to deal with the various orientations of the text in pictures and then manually record the text recognized by Google Goggles.

Not all words in a store are equally effective in the matching process. We observe that words that are *at or above the eye-level*, i.e., towards the ceiling or higher side of the walls, often refer to product categories, menus, banners, etc. These

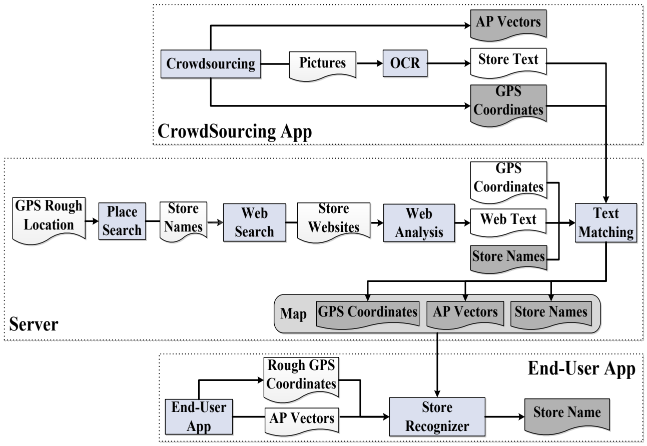

Figure 3: System pipeline of AutoLabel: Crowdsourcing part (top) collects in-store data; Labeling part (middle) processes crowdsourced data to label AP data with store name; End-user part (bottom) utilizes the AP−StoreName mapping to localize users.

words tend to be stable over time and often reflect the core attributes of a store. Given that store webpages are also likely to include these words, we consider at-and-above eye-level words to represent a store. We could extract the words from the desired region in a picture by inferring the attitude of the smartphone camera, from the readings of motion sensors (e.g. gyroscope, accelerometer), when the picture is being taken.

To study the effectiveness of the text *at or above the eye-level*, in Section 3, we evaluate AutoLabel with these two variants of in-store information: i) at or above eye level text only and ii) all text. Apart from the text present in each image, AutoLabel expects that the meta data for the image includes the vector of WiFi APs heard by the device taking the picture.

2.2 Web Data Extraction

Apart from the AP vector, the meta data associated with each in-store image is also expected to include a rough location (could be using GPS). AutoLabel utilizes Google Map to get a list of candidate place names around this rough location. Then, it performs web search with these place names to get the homepages of these businesses and extracts their web text. Besides the text shown on the webpage, many stores also define meta keywords in the html files of their homepages. While the meta keywords are meant for search engines, they usually contain the words which describe some of the key characteristics of that store. So in AutoLabel, we also extract meta keywords from the stores' webpages.

Given that most business web sites have a certain structure, we also leverage it to extract the higher level discriminative words from the web sites. For instance, typical business homepages contain the following parts: Category/Menu section, which is used to categorize the products and navigate to second-layer pages. The menu shown on webpages are usually correlated with the product category shown in the physical stores, because this is how the business owners typically manage and exhibit their products. Therefore, we extract the words from first and second level menus on a web site. To study how this impacts the accuracy of matching the store text and the web text, in Section 3, we evaluate AutoLabel with these two variants: i) menu text only and ii) all web text.

2.3 Matching Text and Labeling APs

The matching process treats the in-store and website text as documents and applies established document matching algorithms, such as Cosine Similarity with *term-frequency inverse document frequency* (TF-IDF) [3] based weights for words. Within a store, words which occur more frequently are likely to be more important than other words in characterizing it. The *term frequency* (TF) method gives more frequent words proportionally higher weight than infrequent ones. On the other hand, if a word appears in only one store in an area, even if infrequently, it helps discriminate that store from others. The *inverse document frequency* (IDF) method captures that intuition by giving higher weight to less common words across the candidate stores.

Specifically, AutoLabel employs TF × IDF as the weight for each word. It uses augmented TF which is computed as

$$\texttt{TF}(t,\mathcal{T}) = 0.5 + \tfrac{0.5\times f(t,\mathcal{T})}{\texttt{maxf}(\mathcal{T})},$$

where $f(t,\mathcal{T})$ is the frequency of word t in text $\mathcal{T}$ (which is the word set of a store's in-store text or web text); $\texttt{maxf}(\mathcal{T})$ is the maximal frequency of the words in $\mathcal{T}$. IDF for a word t depends on its occurrence in the web text of all candidate stores. Suppose n is the total number of candidate web sites and k is the number of sites in which the word t occurs. Then

$$\texttt{IDF}(t) = 1 + log(\tfrac{n}{k}).$$

The resulting weight assigned to a word t in $\mathcal{T}$ is

$$w(t,\mathcal{T}) = \texttt{TF}(t,\mathcal{T}) \times \texttt{IDF}(t).$$

Given the weight of each word, we can compute the Cosine Similarity, SIM_s, between the store text $\mathcal{ST}$ and the web text $\mathcal{WT}_s$ of each candidate store s as

$$SIM_s \leftarrow \frac{\sum_{t\in\mathcal{ST}\bigcap\mathcal{WT}_s} w(t,\mathcal{ST})\times w(t,\mathcal{WT}_s)}{\sqrt{\sum_{t\in\mathcal{ST}} w(t,\mathcal{ST})^2\times\sum_{t\in\mathcal{WT}_s} w(t,\mathcal{WT}_s)^2}}.$$

If the store s^* is the one with the highest similarity measure SIM_{s^*}, then s^* is deemed the name of the store from which the pictures arrived – all WiFi AP vectors from these pictures are labeled with this store name, resulting in a simple WiFiAP−StoreName lookup table.

2.4 Simultaneous Clustering and Labeling

When we do not have an a priori knowledge about the correct clustering of pictures, we can perform simultaneous clustering and labeling as follows. First, we can apply the available WiFi knowledge to gain a crude understanding of pictures that do *not* belong to the same store. As a simple case, two pictures with completely non-overlapping WiFi AP vectors can be assigned to separate clusters. Once we have this basic clustering of pictures, we then form clusters within each of these clusters. Now, in any given iteration of the algorithm, i, we have a set of clusterings, say C_i. We pick each cluster c_{ij} in clustering C_i and compute its matching score against the candidate websites, and eventually sum (c_{ij} $\forall j$) to obtain a score for C_i, say S_i. We repeat this operation for different clusterings of the same pictures. Our intuition is that the correct clustering, say C_r, would achieve the maximum matching score, i.e., $r = \mathrm{argmax}(S_i)$. If this holds, we will be able to label all the stores and their WiFi vectors in one shot, and populate the WiFiAP−StoreName look-up table.

2.5 Semantically Localizing a User

When a user visits a store, her smartphone overhears a WiFi AP vector and performs a lookup on this vector in the WiFiAP–StoreName table. Observe that no store may match exactly with this vector, and many stores may match partially with this vector. We design the vector matching algorithm based on an intuition – we consider the number of APs matched as well as the order of APs, based on their RSSI values, in each of these vectors. We adopt ordered AP vectors approach, as it is observed that such an ordering is fairly stable [4]. The output of this match is a store name that is supplied to apps that provide semantic location based services to the users.

We are currently devising and refining our algorithms for simultaneous clustering and labeling, and also for AP vector matching to semantically localize a user. Hence, in the following section, we limit our evaluation to demonstrate the core idea behind AutoLabel, i.e., matching in-store text and web text to recognize a store.

3. EVALUATION

Before we describe the data collection and present the detailed results, we summarize our findings as follows.

- The similarity between the text from a store and that from any other store, and likewise between the text from a store's website and that from other store's websites, is low, less than 0.3 in 97% of cases.
- In more than 94% of instances, the text from a store matched the most with the text from that store's website than any other store's website.
- In general, around 10 random pictures with text from a store suffice for AutoLabel to distinguish that store from nearby stores with high accuracy.
- Matching the text in a store at or above eye level with the text in the menus of the websites is a reasonable strategy for recognizing a store, though its performance is slightly less than that of using all the available text in the store and on the web.

3.1 Data Collection

We collect data from 18 stores (refer to Fig. 4 for the store names) inside a shopping mall near Champaign, Illinois. Among the 18 stores, some are selling similar merchandise. For instance, two of them are nutrition stores: *GNC* and *Vitamin World*. Three are sports stores (i.e. *Finish Line*, *Foot Locker*, *Mc Sports*), which are selling sports shoes. Several others are clothing stores. Common words could be seen in these similar-business stores. For example, the word *vitamin* occurs in both the nutrition stores.

To accelerate the procedure of taking in-store pictures, we use Google Glass and take panoramic videos inside these stores, which in a way mimics the in-store picture collection by many crowdsourcers over time and provides enough coverage for the store text. The number of pictures with text in each store ranges from 16 to 72. While taking pictures, our crowdsourcers carry smartphones in their pockets, which collect WiFi AP data of the stores they visit. The crowdsourcers also record the semantic names of the places they visit, which serve as the ground truth in evaluating AutoLabel.

3.2 Similarity Between Stores

To distinguish a store from other stores using AutoLabel, it is necessary that: i) the text in those stores are dissimilar; and ii) the text in the websites of those stores are dissimilar. Otherwise, one store's text might match with the web text of two stores, creating ambiguity in recognizing a store. To check whether necessary conditions for AutoLabel hold, we study the similarity between the text of stores in the mall and also the similarity between the text of the websites of these stores.

Fig. 4(left) shows the similarity between the text in stores at the shopping mall. Although the stores with similar business have higher similarity, the overall similarity is low, only 1% of cases have similarity score above 0.3. This gives us confidence that our AutoLabel can distinguish the in-store text for the stores in the same area.

Fig. 4(right) illustrates the similarity of the text in the websites of stores at the mall. It is evident that while the websites of stores having similar businesses have more common text, the overall similarity of web text is low. In 97% of cases, the similarity between the web text of two different stores is less than 0.3, which makes it feasible for AutoLabel to discriminate the web text for the stores in the same area.

3.3 Matching Store Text with Web Text

Next, we study the performance of text matching between the store text and web text. We consider two cases: i) excluding store name and ii) including store name, in case it appears in the text of store images. The intention is to see how well a store can be distinguished based on the text alone without taking advantage of the presence of store name in the text.

Fig. 5 shows the matching result between the store text and web text from 18 stores in the shopping mall. To make the best matching pairs more evident, we plot the similarity scores normalized by the highest score, such that the darker the color is, the better the matching is. Even when the store name is excluded from the text (Fig. 5(left)), the matching score between the store text and the web text of that store is the highest in 16 out of 18 cases, yielding 89% matching accuracy. With store name included, if it happens to appear in the store text (Fig. 5(right)), matching accuracy goes up to 94%.

3.4 Matching with Partial Data

Crowdsourcing is a long-term process. Over time, more useful data gets contributed by more people, and AutoLabel becomes more accurate in correlating the store text with the web text and mapping an AP vector to the store name. In this section, we study how much data is enough for AutoLabel to perform with good accuracy.

3.4.1 Matching with Limited Number of Pictures

To study the accuracy of matching/labeling when different numbers of in-store pictures are available, we randomly select $5{\sim}40$ from all the pictures taken at each store. For each number of pictures, we randomly sample 20 sets. Here, while matching, store name is included in the text if it appears in

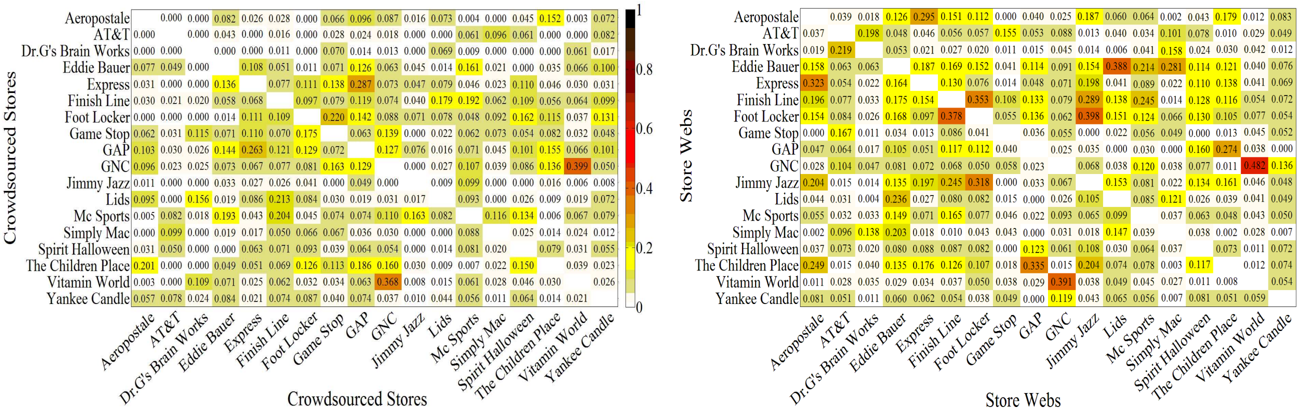

Figure 4: The similarity between: (Left) the text in stores at the shopping mall; (Right) the web text of these stores.

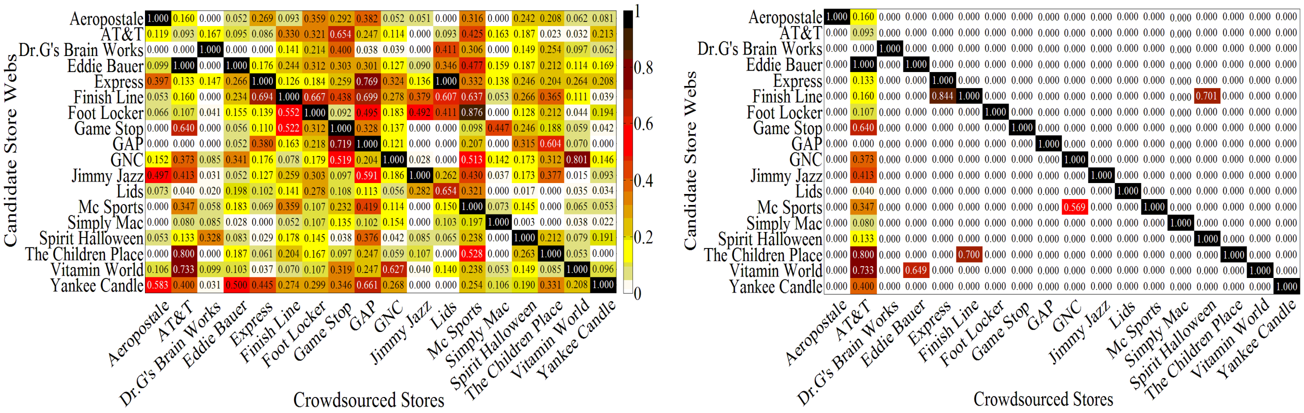

Figure 5: Matching between store text and web text for stores in the mall (similarity scores normalized by the highest similarity value): (Left) excluding and (Right) including the store name if it appears in the store text.

the selected images. Fig. 6 shows the matching accuracy with varying numbers of pictures. It shows that, in many stores, 10 pictures with text allow AutoLabel to build reasonably confident mapping between in-store text and web text. This is because when 10 useful pictures are distributed randomly enough, they should cover the majority of the store. Although more pictures could improve this coverage, the improvement may not be significant.

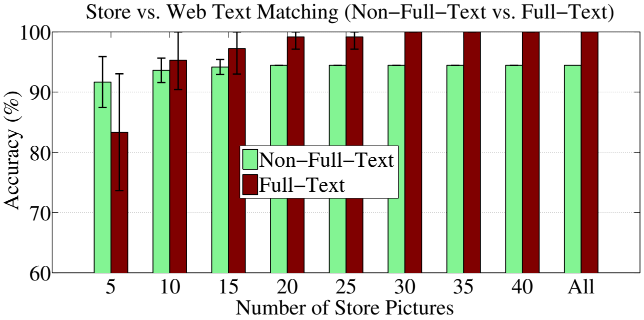

Figure 6: Comparison between non-full-text and full-text strategies with varying numbers of in-store pictures. In almost all scenarios, AutoLabel could correctly match in-store text with the corresponding web text with an average accuracy upwards of 90%.

3.4.2 *Matching with Specific Portion of the Text*

Thus far, all the evaluation of AutoLabel is done matching the text at or above eye-level in the store against the text from menu items of the webpages. An alternative strategy is to utilize all the text in pictures, i.e., including the text below eye-level, and correspondingly, all the text from webpages, including the text in web images. Comparing the performance of these two strategies in Fig. 6, we can see that overall matching accuracy is slightly better using all the text. The only exception is when only a few (e.g. less than 10) store pictures are available, the non-full-text strategy outperforms the full-text strategy. We surmise that when only a few pictures are involved, the non-full-text strategy (which uses text related to menus/categories) is more likely to contain a few but more relevant words, helping it recognize the store better. However, considering that this evaluation is preliminary and that full-text strategy has higher accuracy in general, we need to investigate this further to draw meaningful conclusions.

4. LIMITATIONS AND ON-GOING WORK

While the results of our evaluation are promising, they mainly test the core hypothesis that a store can be recognized by correlating the in-store text with the web text. Towards building an end-to-end AutoLabel system that can facilitate location based services, we are currently implementing and validating algorithms for simultaneous clustering and labeling, and also for AP vector matching to semantically localize a user.

We are also working towards designing a robust store and web text matching algorithm by gaining a better understanding of the hierarchical structure of web pages in terms of product categories and menus, and how that relates to the text on walls, isles and banners within the store. Finally, it is possible that physical stores and webpages also correlate in more subtle dimensions. A holistic understanding of the convergence and divergence of the offline and online worlds is part of our future work.

5. RELATED WORK

The notion of semantic localization is not new. SurroundSense [5] and [6] utilize sensor data from smartphones to characterize the ambience and translate them into a semantic place. However, these approaches still assume labeled ambience data. Authors in [7] attempt to categorize places, by training a model on WiFi, GSM and sensor data collected from frequently visited places. Again, the places are predetermined and pre-labeled.

Authors in [8] utilize GPS data to infer the street address of the place the user visited, and then reverse geo-codes this address to find the place. While this may not be accurate (due to GPS errors indoors), authors correlate with calendar, address book, and even credit card transactions to strengthen precision – a privacy concern. Several other works attempt automatic place identification (e.g., home, office, gym) based on analysis of user trajectories, frequency and timing of visits, and other sensor data [9].

The work in [10] localizes users based on the text signs. It requires users to manually input the text signs they see and matches them against a GPS-tagged sign database. Upon a match, the associated GPS value is returned as the user's location. [11] attempts to identify the stores that appear in a photo by matching it against the images of the exteriors of the nearby stores extracted from the web, a computer vision-heavy proposal. AutoLabel approach is more scalable as it uses the correlation between in-store words and website words to produce a WiFiAP–StoreName table. Moreover, both the above approaches require user intervention to localize self or identify a store from outside, whereas AutoLabel aims to automatically identify the name of the store a user is in.

Perhaps closest to this paper is the work in [12], which tries to connect the text in the crowdsourced pictures with the posts in social networks to infer business names. While similar in spirit, [12] utilizes two disparate information sources: i) crowdsourced pictures, whose content reflects curated choices of the business owner; ii) social-network posts about the business, whose content reflects the unstructured and sometimes unrelated views of the netizens, making the correlation weak – perhaps the reason for its low accuracy of 38%. Also, the authors didn't give any evidence on how and how well they can cluster the crowdsourced pictures. Given that offline version (i.e. the physical store) and online version (i.e. the store's own website) of the same business are curated to have similar look and content, we believe AutoLabel holds better promise as a solution for semantic localization.

6. CONCLUSION

We observe that text – from a physical store and its corresponding website – can serve as an effective bridge between the two worlds. By extracting the words from in-store pictures and correlating with web-words, it is possible to map pictures to their places, ultimately label WiFi MAC addresses with the name of the place. While we are still exploring this idea fully, based on our initial results, we believe this could be a scalable approach, and could lay the essential ground work for physical analytics within and across stores.

7. ACKNOWLEDGEMENTS

We sincerely thank the anonymous reviewers for their invaluable feedback. We are also grateful to Samsung for partially funding this research.

8. REFERENCES

[1] Truc D Le, Thong M Doan, Han N Dinh, and Nam T Nguyen. Isil: Instant search-based indoor localization. In *Consumer Communications and Networking Conference (CCNC)*, pages 143–148. IEEE, 2013.

[2] Google goggles. `www.google.com/mobile/goggles`.

[3] E Garcia. Cosine similarity and term weight tutorial. *Information retrieval intelligence*, 2006.

[4] Qiuxia Chen, Dik-Lun Lee, and Wang-Chien Lee. Rule-based wifi localization methods. In *IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, volume 1, pages 252–258, 2008.

[5] Martin Azizyan, Ionut Constandache, and Romit Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*. ACM, 2009.

[6] Yohan Chon, Nicholas D Lane, Fan Li, Hojung Cha, and Feng Zhao. Automatically characterizing places with opportunistic crowdsensing using smartphones. In *Proceedings of the ACM Conference on Ubiquitous Computing*, pages 481–490, 2012.

[7] Abson Sae-Tang, Michele Catasta, Lucas Kelsey McDowell, and Karl Aberer. Semantic place prediction using mobile data. In *Proceedings of the Mobile Data Challenge Workshop (MDC)*, 2012.

[8] Juhong Liu, Ouri Wolfson, and Huabei Yin. Extracting semantic location from outdoor positioning systems. In *MDM*, 2006.

[9] John Krumm and Dany Rouhana. Placer: semantic place labels from diary data. In *Proceedings of the ACM international joint conference on Pervasive and ubiquitous computing*, pages 163–172, 2013.

[10] Bo Han, Feng Qian, and Moo-Ryong Ra. Human assisted positioning using textual signs. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pages 87–92. ACM, 2015.

[11] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. Visual business recognition: a multimodal approach. In *ACM Multimedia*, pages 665–668, 2013.

[12] Yohan Chon, Yunjong Kim, and Hojung Cha. Autonomous place naming system using opportunistic crowdsensing and knowledge from crowdsourcing. In *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2013.