Distantly Supervised Attribute Detection from Reviews

Lisheng Fu[*]
Computer Science Department
New York University
New York, NY 10003 USA
lisheng@cs.nyu.edu

Pablo Barrio
Google
New York, NY 10011 USA
pjbarrio@google.com

Abstract

This work aims to detect specific attributes of a place (e.g., if it has a romantic atmosphere, or if it offers outdoor seating) from its user reviews via distant supervision: without direct annotation of the review text, we use the crowdsourced attribute labels of the place as labels of the review text. We then use review-level attention to pay more attention to those reviews related to the attributes. The experimental results show that our attention-based model predicts attributes for places from reviews with over 98% accuracy. The attention weights assigned to each review provide explanation of capturing relevant reviews.

1 Introduction

In selecting a product to buy, a restaurant to visit, or a hotel to stay at, people may rely on user reviews but may also filter their choices based on particular attributes (e.g., the availability of an outdoor seating area or the lack of a kid-friendly atmosphere). In limited quantities, these attributes may be collected by hand, but this may be too costly to do on a large scale. So inevitably there will be products for which we have lots of reviews but no attributes. Can these attributes be inferred automatically from the reviews?

We answer this question affirmatively, using restaurant reviews and attributes as our case study. Starting from a large set of reviews and detailed attributes for some of the same restaurants, we train a system to predict the attributes for restaurants for which this information is not available. This is an information extraction task and, as we will show, can be trained through a form of distant supervision.

In our case study, we detect both objective and subjective attributes for restaurants. For objective attributes, we extract detailed facts such as "has outdoor seating". These are fine-grained factual attributes, and differ from entities (e.g., people or locations), which are the focus of entity extraction, a related information extraction task. For subjective attributes, we are detecting fine-grained opinions such as "feels romantic". These are either positive or negative sentiments and not the overall polarity (or rating) in existing sentiment analysis tasks. To the best of our knowledge, we are the first to perform such attribute detection from text.

We propose to address the problem above in a simple distant supervision manner by incorporating an additional data source: We use crowdsourcing to obtain annotation of attributes on places independently, which is much easier to obtain than annotation on the review text. Although the review text of a place does not necessarily indicate the attribute of the place, we hypothesize they are highly correlated and we use the labels on places to be the labels on the review text (Section 2).

As a result, we have a number of reviews for each attribute and place without knowing which review indicates the attribute. We propose to use a review-level attention mechanism to assign high weights to those related reviews. Our experiments show that our simple alignment of the two data sources is effective and the attributes are substantially predictable from the review text. Our best model obtains 98.05% accuracy.

2 Attribute Detection

2.1 Data Sources

Our distant supervision approach takes advantage of two independently created sources of information regarding the restaurants. The first source consists of user reviews written in natural language form with no specific guidance. The second source consists of the labels of predefined at-

[*] This work has been done during the internship at Google.

tributes collected through explicit prompts (a form of crowdsourcing). For instance, a user who has visited the restaurant *Per Se* in New York City may be prompted: "Did *Per Se* offer outdoor seating?". The user can answer the question by selecting one of "*Yes*", "*No*," or "*Not sure*". Due to limited answers to these questions, some restaurants can have both multiple reviews and crowdsourced attribute labels, while many others have only reviews—with no attributes information.
Since attribute labels are crowdsourced, they can be noisy or have disagreement, especially on subjective attributes. In other words, a particular attribute may receive "*Yes*" and "*No*" answers from different users for the same restaurant. We confirm attributes as *Yes* or *No* for a place based on an agreement model that blends the votes and other structured data from the place (e.g., cuisines, location). When the model predicts—with 95% confidence—that at least 2/3 of the voters would respond *Yes*, the model confirms the attribute as *Yes*. We use the same logic for confirming *No*. In this work, we remove the instances where the agreement model is uncertain (i.e., confidence is less than 95%). We use this confident set as ground truth for training and evaluation of our attribute detection model.

2.2 Model Setup

Our goal is to train a model on the restaurants with both reviews and attribute labels and use the model to predict attributes for those with only reviews. The input of the model is the reviews of restaurants and the attributes, while the output of the model is *Yes/No* labels for each attribute. We next describe the basic setup of our neural models, which include an input layer, an encoder, and a decoder.

Input Layer: The input layer consists of *word embedding* and *attribute embedding*. The input layer of the review text is similar to other text classification tasks (e.g., Kim 2014). Each token is converted to a word embedding of dimension d_w. The size of the embedding table is $|V| * d_w$, where $|V|$ is the vocabulary size. For each instance, we look up its attribute embedding A_i from an attribute embedding table. The values of attribute embedding are randomly initialized and trained using backpropagation. The size of the embedding table is $|A| * d_A$, where $|A|$ is the the number of attributes and d_A is the embedding dimension.

Encoder: The encoder reads the word embedding and extracts the feature representation $\phi(x)$ for the bag of reviews, where x is the word tokens of all the reviews. We use a Recurrent Neural Net (RNN) with word-level attention to encode the text of one review. We use GRU (Cho et al., 2014) as the RNN cell. In a single bag, we assume that at least one review will refer to the attribute, while most of the reviews will be unrelated to the attribute. Thus, we use review-level attention on top of the RNN to capture the importance of different reviews. The model will learn high weights for reviews that refer to the attributes and assign almost zero weight to those that are unrelated. This model is similar to the hierarchical attention network (Yang et al., 2016) with two levels of attention.

Given a list of tokens from review text x, we generate a list of word embeddings w_{ij} from the input layer, where i is the index of a review and j is the index of a token in a review. The encoder is defined as the following:

$$h_{ij} = GRU(w_{ij}, h_{ij-1}), \quad (1)$$

$$v_{ij} = tanh(W_v h_{ij} + b_v), \quad (2)$$

$$\alpha_{ij} = \frac{exp(v_{ij}^{\top} v_k)}{\sum_t exp(v_{it}^{\top} v_k)}, \quad (3)$$

$$r_i = \sum_j \alpha_{ij} h_{ij}, \quad (4)$$

$$u_i = tanh(W_u r_i + b_u), \quad (5)$$

$$\alpha_i = \frac{exp(u_i^{\top} u_k)}{\sum_j exp(u_j^{\top} u_k)}, \quad (6)$$

$$\phi(x) = \sum_i \alpha_i r_i, \quad (7)$$

where $W_v, W_u,\ b_v, b_u$ are the weights for the word-level and review-level context vector projections v_{ij} and u_i, respectively. v_k and u_k are the weights of the word-level and review-level context vectors according to the attribute k. r_i is the review embedding computed as the weighted average of h_{ij} according to the importance of the word (α_{ij}) to the attribute k in the review. $\phi(x)$ is the weighted average of r_i according to the importance of the review (α_i) to the attribute k. We refer to this feature representation $\phi(x)$ as a place embedding, since it encodes all the reviews of a place. $\phi(x)$ will in turn be concatenated with an attribute embedding A_i and passed to the decoder.

Decoder: The decoder consists of one hidden layer (h_1) with output label y:

$$h_2 = concat(\phi(x), A_i), \tag{8}$$

$$h_1 = relu(W_2 h_2 + b_2), \tag{9}$$

$$y = W_1 h_1 + b_1, \tag{10}$$

where W_1, W_2, b_1, b_2 are the parameters for the fully connected layers.

3 Experiments

3.1 Dataset

For our case study, we use restaurants and reviews from Google Maps. We constraint the geographic scope of the restaurants to USA and the language of the reviews to English. We can easily extend our dataset to include other categories of places. The crowdsourcing of attribute labels is implemented as a user contribution feature from Google Maps. Those labels include both subjective attributes (e.g., "`feels quiet`") and objective attributes (e.g., "`offering alcohol`"). We choose restaurants with at least 100 reviews to collect enough review text to train the model. In practice, if a place has insufficient reviews, we may not be able to predict attributes based on reviews. Our dataset contains 17k+ of restaurants and 100+ attributes. Each instance consists of one restaurant, one attribute and 100+ reviews. We use 80% of instances for training, 10% for development and 10% for test. We split instances based on restaurants to keep all review text of a single restaurant together and thus avoid overlap between training and evaluation.

3.2 Model Configurations

We use grid search to tune the hyper-parameters. We use 10000 words for the vocabulary size, 100 for the maximum review length (reviews may contain multiple sentences), and 100 for the number of reviews. We use 100 dimensions for both word embedding and attribute embedding. We use 256 filters with window sizes [2,3,4,5] for CNN and 128 states for RNN. We use one hidden layer with 128 units for the decoder. We train the model with the Adam optimizer (Kingma and Ba, 2014) and use cross entropy as the loss function. Our learning rate is set to 0.001 and our batch size is set to 32.

3.3 Predictability from Review Text

Since the model is doing binary prediction (i.e., the model predicts *Yes* or *No* for an attribute), we use accuracy as our quality metric. It evaluates the label prediction of an attribute-restaurant pair (one instance). Since we do not have labels of all attributes for every restaurant, we do not report accuracy by restaurant.

We compare against multiple baselines to show the effectiveness of our model choice.

- **Majority**: it predicts the most frequent label for an attribute. This is equivalent to using attribute embedding alone to train the model without reviews.
- **BoW**: it uses the average of the word embeddings as the review embedding, and then uses the average of review embedding as the place embedding.
- **CNN**: it uses CNN to extract the feature representation from the word embedding as the review embedding.
- **CNN + RATT**: it uses review-level attention (RATT) to construct the place embedding from the review embedding instead of taking average.
- **RNN**: it uses RNN to extract the feature representation from the word embedding. The difference from our proposed model is that it does not use the review-level attention.

The input and decoder are the same for these models. The Majority baseline obtains 90.82% accuracy, which indicates the label bias in the dataset. The label bias is intrinsic for some attributes, and possibly increases after uncertain instances are removed from the dataset by the agreement model. More sophisticated models can perform much better with better sentence understanding ability (see Table 1). We observe continuous improvement with more capable encoders of sentence. Moreover, attention-based aggregation for reviews further improves the accuracy for both CNN and RNN models.

As shown, our RNN+RATT model performs the best, yielding 98.05% accuracy. This indicates that review text can highly predict the presence or not of an attribute. It is also very impressive that such high accuracy is obtained via distant supervision (i.e., without direct annotation on text).

Model	Accuracy
Majority	90.82
BOW	96.48
CNN	97.17
CNN+RATT	97.37
RNN	97.84
RNN+RATT	**98.05**

Table 1: Model accuracy.

Model	A_1	A_2	A_3
Majority	66.00	51.00	71.09
BoW	82.58	74.77	87.39
CNN	86.32	78.60	91.55
CNN+RATT	88.39	85.05	91.43
RNN	95.35	82.75	93.88
RNN+RATT	**96.65**	**85.81**	**94.25**

Table 2: Accuracy for some ambiguous attributes. A_1: usually a wait, A_2: has outdoor seating, A_3: serves late night food.

This confirms our hypothesis that the review text should contain the knowledge of attributes and is probably effective to predict attributes.

In the dataset, there is a substantial fraction of the attributes that are nearly always either positive or negative across places. As an example, consider that most restaurants can accept "`pay by credit card`" and are rarely "`cash only`" These attributes are relatively easy to predict, which causes the overall accuracy to be high. There are also attributes that are harder for simple models to predict. For those attributes, the sophisticated models works significantly better than the baselines (Table 2). We also observe improvement by adding the review-level attention. This verifies our hypothesis that giving more weights to relevant reviews could help since we align labels to a bag of reviews of a place without knowing which review indicates the attribute.

We next show some examples to explain how the review-level attention works. (Table 3). It often captures the important one out of all the reviews in a place (e.g. the first three examples), but sometimes fails because of the misleading keyword (e.g. "2 hr" in the fourth example). There are also cases where reviews may not tell anything about the attribute (e.g. the fifth example), which is hard to avoid when we use labels not directly annotated from text. Fortunately, this does not often occur in the dataset. The sixth example indicates the case where the attended review does show related information about the attribute, but not enough to conclude. The model might have combined several reviews to draw the conclusion in this case.

4 Related Work

The idea of distant supervision has been proposed and used widely in Relation Extraction (Mintz et al., 2009; Riedel et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012), where the source of labels is an external knowledge base. The label assignment is done via aligning entities from knowledge base to text. In alignment, relation extraction has the problem that not every entity pair expresses the semantic relation stored in the knowledge base. We can view our crowdsourced attribute labels as a knowledge base of places and their attributes. The label alignment in our case is much simpler, since both attributes and reviews are associated with the place. The review text, on the other hand, may or may not express the attribute acquired from crowdsourcing. Recently (Lin et al., 2016) used neural methods to achieve state-of-the-art for distantly supervised relation extraction. We thus focus on neural methods in our modeling.

The attribute detection task is also similar to the aspect-based sentiment analysis task (Pontiki et al., 2016), but contains both subjective and objective aspects. We take a completely different approach in this paper to tackle the problem by using distant supervision and create significantly larger amount of the training data. It might be an interesting direction to use this distant supervision way to create more training data for the aspect-based sentiment analysis.

5 Conclusion and Future Work

We attempt to detect specific attributes of places using two sources of data: the review text of places and their crowdsourced attribute labels. We create training data from the two sources in a form of distant supervision. We use a review-level attention mechanism to pay attention to reviews related to the attribute. From the experimental results, we find that the review text is highly predictive of the attributes despite the lack of shared guidance during generation of two sources of data. Our method requires no direct annotation on text, which will

Attribute	L	P	A	Review Text	Notes
usually a wait	Y	Y	0.08	... Just be prepared to wait or otherwise get lucky and find a seat at the bar ! ...	Missed by BoW
has outdoor seating	Y	Y	0.17	If you want to eat in front plan on waiting after signing up to the list on busy mornings , but the back patio is just as nice ...	Missed by BoW
requires cash only	N	Y	0.11	... Remember they are Cash Only !	Wrong label
usually a wait	N	Y	0.06	Got there after 2 hr drive and found the owners on vacation and the place closed ...	Irrelevant
pay by credit card	Y	Y	0.25	Food and service is great Tanisha is a awesome sever	No related review
usually a wait	Y	Y	0.09	...Never a long wait for to go orders...	Tricky

Table 3: Attributes along with true label (L), prediction (P), review-level attention weight (A), and review text.

make attribute detection more feasible in practice.

In creating the crowdsourced labels, we use an agreement model to select most agreed labels for attributes. It will be interesting future work to extend this to raw user votes. We will have a more realistic dataset, especially for subjective attributes where users may have conflict opinions.

References

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoderdecoder for statistical machine translation. In *Proceedings of EMNLP*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of ACL*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Luan Huanbo, and Sun Maosong. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL*.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 19–30.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of ECML*.

Mihai Surdeanu, Julie Tibshirani Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of EMNLP*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL*.