

Learning Document-Level Semantic Properties from Free-text Annotations

S. R. K. Branavan and Harr Chen and Jacob Eisenstein and Regina Barzilay

Computer Science and Artificial Intelligence Laboratory

Massachusetts Institute of Technology

77 Massachusetts Ave., Cambridge MA 02139

{branavan, harr, jacob, regina}@csail.mit.edu

Abstract

This paper demonstrates a new method for leveraging unstructured annotations to infer semantic document properties. We consider the domain of product reviews, which are often annotated by their authors with free-text keyphrases, such as “a real bargain” or “good value.” We leverage these unstructured annotations by clustering them into semantic properties, and then tying the induced clusters to hidden topics in the document text. This allows us to predict relevant properties of unannotated documents. Our approach is implemented in a hierarchical Bayesian model with joint inference, which increases the robustness of the keyphrase clustering and encourages document topics to correlate with semantically meaningful properties. We perform several evaluations of our model, and find that it substantially outperforms alternative approaches.

1 Introduction

A central problem in language understanding is transforming raw text into structured representations. Learning-based approaches have dramatically increased the scope and robustness of automatic language processing, but they are typically dependent on large expert-annotated datasets, which are costly to produce. In this paper, we show how novice-generated free-text annotations available online can be leveraged to automatically infer document-level semantic properties.

More concretely, we are interested in determining properties of consumer products and services

pros/cons: <i>great nutritional value</i> ... combines it all: an amazing product, <i>quick and friendly service</i> , cleanliness, great nutrition ...
pros/cons: <i>a bit pricey, healthy</i> ... is an awesome place to go if you are health conscious. They have some really great low calorie dishes and they publish the calories and fat grams per serving.

Figure 1: Excerpts from online restaurant reviews with pros/cons phrase lists. Both reviews discuss healthiness, but use different keyphrases.

from reviews. Often, such reviews are annotated with *keyphrase* lists of pros and cons. We would like to use these keyphrase lists as training labels, so that the properties of unannotated reviews can be predicted. However, novice-generated keyphrases lack consistency: the same underlying property may be expressed many ways, *e.g.*, “reasonably priced” and “a great bargain.” To take advantage of such noisy labels, a system must both uncover their hidden *clustering* into properties, and learn to predict these properties from review text.

This paper presents a model that attacks both problems simultaneously. We assume that both the review text and the selection of keyphrases are governed by the underlying hidden properties of the review. Each property indexes a language model, thus allowing reviews that incorporate the same property to share similar features. In addition, each observed keyphrase is associated with a property; keyphrases that are associated with the same property should have similar distributional and surface features.

We link these two ideas in a joint hierarchical

Bayesian model. Keyphrases are clustered based on their distributional and orthographic properties, and a hidden topic model is applied to the review text. Crucially, the keyphrase clusters and document hidden topics are linked, and inference is performed jointly. This increases the robustness of the keyphrase clustering, and ensures that the inferred hidden topics are indicative of salient semantic properties.

Our method is applied to a collection of reviews in two distinct categories: restaurants and cell phones. During training, lists of keyphrases are included as part of the reviews by the review authors. We then evaluate the ability of our model to predict review properties when the keyphrase list is hidden. Across a variety of evaluation scenarios, our algorithm consistently outperforms alternative strategies by a wide margin.

2 Related Work

Review Analysis Our approach relates to previous work on property extraction from reviews (Popescu et al., 2005; Hu and Liu, 2004; Kim and Hovy, 2006). These methods extract lists of phrases, which are analogous to the keyphrases we use as input to our algorithm. They operate using manually compiled sets of rules or machine learning approaches. Our work is distinguished in two ways. First, we are able to predict keyphrases beyond those that appear verbatim in the text. Second, our approach also learns the relationships between different keyphrases, allowing us to draw direct comparisons between reviews.

Bayesian Topic Modeling One aspect of our model views properties as distributions over words in the document. This approach is inspired by methods in the topic modeling literature, such as Latent Dirichlet Allocation (Blei et al., 2003), where topics are treated as hidden variables that govern the distribution of words in a text. Our algorithm extends this notion by biasing the induced hidden topics toward a clustering of known keyphrases. Tying these two information sources together enhances the robustness of the hidden topics, thereby increasing the chance that the induced structure corresponds to semantically meaningful properties.

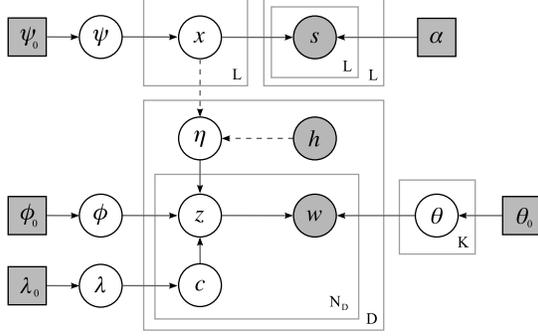
3 Problem Formulation

We formulate our problem as follows. We assume a dataset composed of documents with associated keyphrases. Each document may be marked with multiple keyphrases that express semantic properties. Across the entire collection, several keyphrases may express the same property. The keyphrases are also incomplete – review texts often express properties that are not mentioned in their keyphrases. At training time, our model has access to both text and keyphrases; at test time, the goal is to predict which properties a previously unseen document supports, and by extension, which keyphrases are applicable to it.

4 Model Description

Our approach leverages both keyphrase clustering and distributional analysis of the text in a joint, hierarchical Bayesian model. Keyphrases are drawn from a set of clusters; words in the documents are drawn from language models indexed by a set of topics, where the topics correspond to the keyphrase clusters. Crucially, we bias the assignment of hidden topics in the text to be similar to the topics represented by the keyphrases of the document, but we permit some words to be drawn from other topics not represented by the document’s keyphrases. This flexibility in the coupling allows the model to learn effectively in the presence of incomplete keyphrase annotations, while still encouraging the keyphrase clustering to cohere with the topics supported by the document text. The plate diagram for our model is shown in Figure 2.

We train the model on documents annotated with keyphrases. During training, we learn a hidden topic model from the text; each topic is also associated with a cluster of keyphrases. At test time, we are presented with documents that do not contain keyphrase annotations. The hidden topic model of the review text is used to determine the properties that a document as a whole supports. For each property, we compute the proportion of the document’s words assigned to it. Properties with proportions above a set threshold (tuned on a development set) are predicted as being supported.



- ψ – keyphrase cluster model
- x – keyphrase cluster assignment
- s – keyphrase similarity values
- h – document keyphrases
- η – document keyphrase topics
- λ – probability of selecting η instead of ϕ
- c – selects between η and ϕ for word topics
- ϕ – background word topic model
- z – word topic assignment
- θ – language models of each topic
- w – document words

$$\begin{aligned} \psi &\sim \text{Dirichlet}(\psi_0) \\ x_\ell &\sim \text{Multinomial}(\psi) \\ s_{\ell, \ell'} &\sim \begin{cases} \text{Beta}(\alpha_{=}) & \text{if } x_\ell = x_{\ell'} \\ \text{Beta}(\alpha_{\neq}) & \text{otherwise} \end{cases} \\ \eta_d &= [\eta_{d,1} \dots \eta_{d,K}]^T \\ &\text{where } \eta_{d,k} \propto \begin{cases} 1 & \text{if } x_\ell = k \text{ for any } \ell \in h_d \\ 0 & \text{otherwise} \end{cases} \\ \lambda &\sim \text{Beta}(\lambda_0) \\ c_{d,n} &\sim \text{Bernoulli}(\lambda) \\ \phi &\sim \text{Dirichlet}(\phi_0) \\ z_{d,n} &\sim \begin{cases} \text{Multinomial}(\eta_d) & \text{if } c_{d,n} = 1 \\ \text{Multinomial}(\phi) & \text{otherwise} \end{cases} \\ \theta_k &\sim \text{Dirichlet}(\theta_0) \\ w_{d,n} &\sim \text{Multinomial}(\theta_{z_{d,n}}) \end{aligned}$$

Figure 2: The plate diagram for our model. Shaded circles denote observed variables, and squares denote hyperparameters. The dotted arrows indicate that η is constructed deterministically from \mathbf{x} and \mathbf{h} .

4.1 Keyphrase Clustering

One of our goals is to cluster the keyphrases, such that each cluster corresponds to a well-defined document property. While our overall model is generative, we desire the freedom to use any arbitrary metric for keyphrase similarity. For this reason, we represent each distinct keyphrase as a vector of similarity scores computed over the set of observed keyphrases; these scores are represented by s in Figure 2. We then explicitly generate this similarity matrix, rather than the surface form of the keyphrase itself. Modeling similarity scores rather than keyphrase words affords us the flexibility of clustering the keyphrases using more than just their word distributions. We assume that similarity scores are conditionally independent given the keyphrase clustering. Models that make similar assumptions about the independence of related hidden variables have been shown to be successful (Toutanova and Johnson, 2007), though this is an area of future work for us.

Similarity between keyphrases is computed using

a linear interpolation of two metrics. The first is the cosine similarity between keyphrase word vectors. The second is based on the co-occurrence of keyphrases in the review texts themselves. While we chose these two metrics for their simplicity, our model is inherently capable of using other sources of similarity information. For a discussion of similarity metrics, see (Lin, 1998).

4.2 Document-level Distributional Analysis

Our analysis of the document text is based on probabilistic topic models such as LDA (Blei et al., 2003). In the LDA framework, each word is generated from a language model that is indexed by the word’s topic assignment. Thus, rather than identifying a single topic for a document, LDA identifies a distribution over topics.

Our word model operates similarly, identifying a topic for each word, written as z in Figure 2. However, where LDA learns a distribution over topics for each document, we deterministically construct a document-specific topic distribution from the clusters represented by the document’s keyphrases – this

is η in the figure. η assigns equal probability to all topics that are represented in the keyphrases, and zero probability to other topics. Generating the word topics in this way ties together the keyphrase clustering and language models.

As noted above, sometimes properties are expressed in the text even when no related keyphrase is present. For this reason, we also construct another “background” distribution ϕ over topics, which is shared across documents. The auxiliary variable c indicates whether a given word’s topic is drawn from the set of keyphrase clusters, or from the background model.

4.3 Generative Process

In this section, we describe the underlying generative process more formally.

First we consider the set of all keyphrases observed across the entire corpus, of which there are L . We draw a multinomial distribution ψ over the K keyphrase clusters from a symmetric Dirichlet prior ψ_0 . Then for the ℓ^{th} keyphrase, a cluster assignment x_ℓ is drawn from the multinomial ψ . Finally, the similarity matrix $\mathbf{s} \in [0, 1]^{L \times L}$ is constructed. Each entry $s_{\ell, \ell'}$ is drawn independently, depending on the cluster assignments x_ℓ and $x_{\ell'}$. Specifically, $s_{\ell, \ell'}$ is drawn from a Beta distribution with parameters $\alpha_ =$ if $x_\ell = x_{\ell'}$ and α_{\neq} otherwise. The parameters $\alpha_ =$ linearly bias $s_{\ell, \ell'}$ towards one (Beta($\alpha_ =$) \equiv Beta(2, 1)), and the parameters α_{\neq} linearly bias $s_{\ell, \ell'}$ towards zero (Beta(α_{\neq}) \equiv Beta(1, 2)).

Next, the words in each of the D documents are generated. Document d has N_d words, and the topic for word $w_{d,n}$ is written as $z_{d,n}$. These latent topics are drawn either from the set of clusters represented in the document’s keyphrases, or from a background topic model ϕ . We deterministically construct a document-specific keyphrase topic model η , based on the keyphrase cluster assignments \mathbf{x} and the observed keyphrases \mathbf{h} . The multinomial η_d assigns equal probability to each topic that is represented by a phrase in h_d , and zero probability to other topics.

As noted earlier, a document’s text may support properties that are not mentioned in its observed keyphrases. For that reason, we draw a background topic multinomial ϕ from a symmetric Dirichlet prior ϕ_0 . The binary auxiliary variable $c_{d,n}$ determines whether the word’s topic is drawn from the

keyphrase model η_d or the background model ϕ . $c_{d,n}$ is drawn from a weighted coin flip, with probability λ ; λ is drawn from a Beta distribution with prior λ_0 . We have $z_{d,n} \sim \eta_d$ if $c_{d,n} = 1$, and $z_{d,n} \sim \phi$ otherwise. Finally, the word $w_{d,n}$ is drawn from the multinomial $\theta_{z_{d,n}}$, where $z_{d,n}$ indexes a topic-specific language model. Each of the K language models θ_k is drawn from a symmetric Dirichlet prior θ_0 .

5 Posterior Sampling

Ultimately, we need to compute the model’s posterior distribution given the training data. Doing so analytically is intractable due to the complexity of the model. In these cases, standard sampling techniques can be used to estimate the posterior. Our model lends itself to estimation via a straightforward Gibbs sampler, one of the more commonly used and simpler approaches to sampling.

By computing conditional distributions for each hidden variable given the other variables, and repeatedly sampling each of these distribution in turn, we can build a Markov chain whose stationary distribution is the posterior of the model parameters (Gelman et al., 2004). Other work in NLP that employs sampling techniques includes (Finkel et al., 2005; Goldwater et al., 2006). We now present sampling equations for each of the hidden variables in Figure 2.

The prior over keyphrase clusters ψ is sampled based on hyperprior ψ_0 and keyphrase cluster assignments \mathbf{x} . We write $p(\psi \mid \dots)$ to mean the probability conditioned on all the other variables.

$$\begin{aligned} p(\psi \mid \dots) &\propto p(\psi \mid \psi_0) p(\mathbf{x} \mid \psi), \\ &= p(\psi \mid \psi_0) \prod_{\ell} p(x_{\ell} \mid \psi) \\ &= \text{Dir}(\psi; \psi_0) \prod_{\ell} \text{Mul}(x_{\ell}; \psi) \\ &= \text{Dir}(\psi; \psi'), \end{aligned}$$

where ψ'_i is $\psi_0 + \text{count}(x_{\ell} = i)$. This update rule is due to the conjugacy of the multinomial to the Dirichlet distribution. The first line follows from Bayes’ rule, and the second line from the conditional independence of similarity scores \mathbf{s} given \mathbf{x} and α , and of word topic assignments \mathbf{z} given η , ψ , and \mathbf{c} .

$$\begin{aligned}
p(x_\ell | \dots) &\propto p(x_\ell | \psi) p(\mathbf{s} | x_\ell, \mathbf{x}_{-\ell}, \alpha) p(\mathbf{z} | \eta, \psi, \mathbf{c}) \\
&\propto p(x_\ell | \psi) \left[\prod_{\ell' \neq \ell} p(s_{\ell, \ell'} | x_\ell, x_{\ell'}, \alpha) \right] \left[\prod_d \prod_{c_{d,n}=1} p(z_{d,n} | \eta_d) \right] \\
&= \text{Mul}(x_\ell; \psi) \left[\prod_{\ell' \neq \ell} \text{Beta}(s_{\ell, \ell'}; \alpha_{x_\ell, x_{\ell'}}) \right] \left[\prod_d \prod_{c_{d,n}=1} \text{Mul}(z_{d,n}; \eta_d) \right]
\end{aligned}$$

Figure 3: The resampling equation for the keyphrase cluster assignments.

Resampling equations for ϕ and θ_k can be derived in a similar manner:

$$\begin{aligned}
p(\phi | \dots) &\propto \text{Dir}(\phi; \phi'), \\
p(\theta_k | \dots) &\propto \text{Dir}(\theta_k; \theta_{k'}),
\end{aligned}$$

where $\phi'_i = \phi_0 + \text{count}(z_{n,d} = i \wedge c_{n,d} = 0)$ and $\theta'_{k,i} = \theta_0 + \text{count}(w_{n,d} = i \wedge z_{n,d} = k)$. In building the counts for ϕ'_i , we consider only cases in which $c_{n,d} = 0$, indicating that the topic $z_{n,d}$ is indeed drawn from the background topic model ϕ . Similarly, when building the counts for θ'_k , we consider only cases in which the word $w_{d,n}$ is drawn from topic k .

To resample λ , we employ the conjugacy of the Beta prior to the Bernoulli observation likelihoods, adding counts of c to the prior λ_0 .

$$p(\lambda | \dots) \propto \text{Beta}(\lambda; \lambda'),$$

$$\text{where } \lambda' = \lambda_0 + \begin{bmatrix} \text{count}(c_{d,n} = 1) \\ \text{count}(c_{d,n} = 0) \end{bmatrix}.$$

The keyphrase cluster assignments are represented by \mathbf{x} , whose sampling distribution depends on ψ , \mathbf{s} , and \mathbf{z} , via η . The equation is shown in Figure 3. The first term is the prior on x_ℓ . The second term encodes the dependence of the similarity matrix \mathbf{s} on the cluster assignments; with slight abuse of notation, we write $\alpha_{x_\ell, x_{\ell'}}$ to denote $\alpha_{=}$ if $x_\ell = x_{\ell'}$, and α_{\neq} otherwise. The third term is the dependence of the word topics $z_{d,n}$ on the topic distribution η_d . We compute the final result of Figure 3 for each possible setting of x_ℓ , and then sample from the normalized multinomial.

The word topics \mathbf{z} are sampled according to the topic distribution η_d , the background distribution ϕ ,

the observed words \mathbf{w} , and the auxiliary variable \mathbf{c} :

$$\begin{aligned}
p(z_{d,n} | \dots) &\propto p(z_{d,n} | \phi, \eta_d, c_{d,n}) p(w_{d,n} | z_{d,n}, \theta) \\
&= \begin{cases} \text{Mul}(z_{d,n}; \eta_d) \text{Mul}(w_{d,n}; \theta_{z_{d,n}}) & \text{if } c_{d,n} = 1 \\ \text{Mul}(z_{d,n}; \phi) \text{Mul}(w_{d,n}; \theta_{z_{d,n}}) & \text{otherwise.} \end{cases}
\end{aligned}$$

As with x , each $z_{d,n}$ is sampled by computing the conditional likelihood of each possible setting within a constant of proportionality, and then sampling from the normalized multinomial.

Finally, we sample the auxiliary variables $c_{d,n}$, which indicates whether the hidden topic $z_{d,n}$ is drawn from η_d or ϕ . \mathbf{c} depends on its prior λ and the hidden topic assignments \mathbf{z} :

$$\begin{aligned}
p(c_{d,n} | \dots) &\propto p(c_{d,n} | \lambda) p(z_{d,n} | \eta_d, \phi, c_{d,n}) \\
&= \begin{cases} \text{Bern}(c_{d,n}; \lambda) \text{Mul}(z_{d,n}; \eta_d) & \text{if } c_{d,n} = 1 \\ \text{Bern}(c_{d,n}; \lambda) \text{Mul}(z_{d,n}; \phi) & \text{otherwise.} \end{cases}
\end{aligned}$$

Again, we compute the likelihood of $c_{d,n} = 0$ and $c_{d,n} = 1$ within a constant of proportionality, and then sample from the normalized Bernoulli distribution.

6 Experimental Setup

Data Sets We evaluate our system on reviews from two categories, restaurants and cell phones. These reviews were downloaded from the popular Epinions¹ website. Users of this website evaluate products by providing both a textual description of their opinion, as well as concise lists of keyphrases (pros

¹<http://www.epinions.com/>

and cons) summarizing the review. The statistics of this dataset are provided in Table 1. For each of the categories, we randomly selected 50%, 15%, and 35% of the documents as training, development, and test sets, respectively.

Manual analysis of this data reveals that authors often omit properties from the list of keyphrases that are mentioned in the text. To obtain a complete gold standard, we annotated a subset of the reviews from the restaurant category manually. The annotation effort focused on eight properties that were commonly mentioned by the authors. These included properties underlying keyphrases such as “pleasant atmosphere” and “attentive staff.” Two annotators performed this task, annotating collectively 160 reviews. 30 reviews were annotated by both. The Cohen’s kappa, a measure of interannotator agreement that ranges from zero to one, is 0.78 on this joint set, indicating high agreement (Cohen, 1960). Each review was annotated with 2.56 properties on average.

	Restaurants	Cell Phones
# of reviews	3883	1112
avg. review length	916.9	1056.9
avg. keyphrases / review	3.42	4.91

Table 1: Statistics of the reviews dataset by category.

Training Details Our model needs to be provided with the number of clusters K . We set K large enough for the model to learn effectively on the development set. For example, in the restaurant category, where the gold standard has eight clusters, we set K to 20. In the cell phone category, it was set to 30.

As mentioned before, we use Gibbs sampling to estimate the parameters of our model. To improve the model’s convergence rate, we perform two initialization steps. In the first step, Gibbs sampling is done only on the keyphrase clustering component of the model, ignoring document text. The second step fixes this keyphrase clustering and samples the rest of the parameters in the model. These initialization steps are run for 5,000 iterations each. The full joint model is then sampled for 10,000 iterations. Inspection of the parameter estimates confirms model convergence. On a 2GHz single-core desktop machine, model training as implemented in Matlab takes about two hours.

The final point estimate used for testing is an average (for continuous variables) or a mode (for discrete variables) over the last 1,000 Gibbs sampling iterations. Averaging is a heuristic that is applicable in our case because our sample histograms are unimodal and exhibit low skew. The model usually works equally well using one-sample estimates, but is more prone to estimation noise.

As previously mentioned, we convert word topic assignments to document properties by examining the proportion of words supporting each property. A proportion threshold is set for each property via the development set.

Evaluation Procedures Our first evaluation examines the accuracy of our model and the baselines by comparing their output against the keyphrases provided by the review authors. More specifically, we test whether the model supports each of the author’s actual keyphrases, given the review.

As mentioned before, the author’s keyphrases are incomplete. Therefore to perform a noise-free comparison, we based our second evaluation on the manually constructed gold standard for the restaurant category. We took the most commonly observed keyphrase from each of the eight annotated properties, and tested whether the model supports them.

In both types of evaluation, we measure the model’s performance using precision, recall, and F-score. These are computed in the standard manner, based on the model’s keyphrase predictions compared against the corresponding references. The sign test was used for statistical significance testing.

Baselines To the best of our knowledge the task of simultaneously identifying and predicting multiple properties has not been addressed in the literature. We therefore consider five baselines that allow us to explore the properties of this task and our model.

Random: Each keyphrase is supported by a document with probability of one half. This baseline’s results are computed (in expectation) rather than actually run. This method is expected to have a recall of 0.5, because in expectation it will select half of the correct keyphrases. Its precision is the proportion of supported keyphrases in the test set.

Phrase in text: A keyphrase is supported by a document if it appears verbatim in the text. Precision should be high whereas recall will be low, because of the strict requirements for a keyphrase to be sup-

	Restaurants gold annotation			Restaurants free-text annotation			Cell phones free-text annotation		
	Recall	Precis.	F-Score	Recall	Precis.	F-Score	Recall	Precis.	F-Score
Random baseline	0.5000	0.3000	* 0.3750	0.5000	0.5000	* 0.5000	0.5000	0.4886	* 0.4943
Phrase in text	0.0443	0.4828	* 0.0812	0.0779	0.9091	* 0.1435	0.1524	0.6400	* 0.2462
Cluster in text	0.2880	0.3583	◇ 0.3193	0.5247	0.6433	* 0.5780	0.6952	0.5448	◇ 0.6109
Phrase classifier	0.0222	0.6364	* 0.0428	0.0675	0.9630	* 0.1262	0.0190	0.6667	* 0.0370
Cluster classifier	0.0981	0.4769	* 0.1627	0.2286	0.8980	* 0.3644	0.1714	0.8182	0.2835
Our model	0.6076	0.3879	0.4735	0.7439	0.7073	0.7251	0.6762	0.6174	0.6455

Table 2: Comparison of the property predictions made by our model and the baselines in the two categories as evaluated against the gold and free-text annotations. The methods against which our model has significantly better results on the sign test are indicated with a * for $p \leq 0.05$, and \diamond for $p \leq 0.1$

	clustering	Restaurants			Cell phones		
		Recall	Precision	F-Score	Recall	Precision	F-Score
Cluster in text	automatic	0.5247	0.6433	* 0.5780	0.6952	0.5448	0.6109
	gold	0.5429	0.6076	* 0.5734	0.9143	0.4974	0.6443
Cluster classifier	automatic	0.2286	0.8980	0.3644	0.1714	0.8182	0.2835
	gold	0.2208	0.9043	0.3549	0.1619	0.7391	0.2656
Our model	automatic	0.7439	0.7073	0.7251	0.6762	0.6174	0.6455
	gold	0.7195	0.7084	0.7139	0.7238	0.5547	0.6281

Table 3: Our model and two of the baselines make use of paraphrasing information derived from our model’s clustering. By providing these methods with the gold standard clustering instead, we can indirectly evaluate the quality of our model’s clustering, and its impact on inference. Using the gold clustering gives a statistically insignificant difference from the model’s clustering, except for the pair indicated by *, where the gold clustering actually performed worse.

ported.

Cluster in text: A keyphrase is supported by a document if it or any of its paraphrases appears in the text. Paraphrasing is based on our model’s clustering of the keyphrases. The use of paraphrasing information enhances recall at the potential cost of precision, depending on the quality of the clustering.

Phrase classifier: A separate discriminative classifier is trained for each keyphrase. Positive examples are documents that are labeled by the author with the keyphrase; all other documents are negative examples. A keyphrase is supported by a document if that keyphrase’s classifier returns positive.

Cluster classifier: A separate discriminative classifier is trained for each cluster of keyphrases. Positive examples are documents that are labeled by the author with any keyphrase from the cluster; all other documents are negative examples. All keyphrases of a cluster are supported by a document if that cluster’s classifier returns positive. Keyphrase clustering is based on our model.

Phrase classifier and *cluster classifier* employ maximum entropy classifiers, trained on the same

features as our model, *i.e.*, word counts. As with the last two baselines, the former is high-precision/low-recall, because for any particular keyphrase, its synonymous keyphrases would be considered negative examples. The latter broadens the positive examples, improving recall while likely hurting precision. We used Zhang Le’s Maxent toolkit² to build these classifiers.

7 Results

Table 2 presents the results of the two evaluation scenarios described above. Our model outperforms every baseline by a wide margin in all evaluations.

The absolute performance of the automatic methods indicates the difficulty of the task. For instance, evaluation against gold annotations (see Table 2) shows that the random baseline outperforms all of the other baselines. We observe similarly disappointing results for the baselines on the restaurant category against the free-text annotations. The precision and recall characteristics of the baselines

²http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

match our previously described expectations.

The poor performance of the discriminative models seems surprising at first. However, these results can be explained by the degree of noise in the training data, specifically, the aforementioned sparsity of free-text annotations. As previously described, our technique allows document text topics to stochastically derive from either the keyphrases or a background distribution – this allows our model to learn effectively from incomplete annotations. In fact, when we force all text topics to derive from keyphrase clusters in our model, its performance degrades to the level of the classifiers or below, with an F-score of 0.3900 in the restaurant category and 0.1714 in the cell phone category (compare to free-text results in Table 2).

As expected, paraphrasing information contributes significantly to baseline performance, generally improving recall with low impact on precision. In fact, in some instances adding paraphrasing information to the *phrase in text* baseline raises its performance to a level close to that of our model’s. As previously observed in entailment research (Dagan et al., 2006), paraphrasing information contributes greatly to improved performance in inference tasks.

In light of this observation, it is important to quantify the quality of automatically computed paraphrases. One way to assess clustering quality is to compare it against a “gold standard” clustering, as constructed by humans. For this purpose, we use the *Rand Index* (Rand, 1971), a measure of cluster similarity. This measure varies from zero to one; higher scores are better. In the restaurant category, the Rand Index of our model’s clusters is 0.9441; for cell phones, it is 0.9086.

Another way of assessing cluster quality is to consider the impact of using the gold clustering instead of our model’s clustering in our model and the *cluster in text* and *cluster classifier* baselines. As Table 3 shows, using the model clustering yields results comparable to using the gold clustering. This indicates that for the purposes of our task, the model clustering is of sufficient quality.

8 Conclusions and Future Work

In this paper, we have shown how free-text annotations provided by novice users can be leveraged as a

training set for document-level semantic inference. The resulting system overcomes the lack of consistency in such annotations by inducing a hidden structure of semantic properties, which correspond both to clusters of keyphrases and hidden topic models in the text. Our approach takes the form of a hierarchical Bayesian model, and straightforward inference is possible using Gibbs sampling. The resulting system successfully extracts semantic properties of unannotated restaurant and cell phone reviews, empirically validating our approach.

We see multiple avenues of future work. First, our model draws substantial power from features that measure keyphrase similarity. This ability to use arbitrary similarity metrics is desirable; however, representing individual similarity scores as random variables is a compromise, as they are clearly not independent. We believe that this problem could be avoided by modeling the generation of the entire similarity matrix jointly.

We have assumed that the properties themselves are essentially unstructured. In reality, properties are related in interesting ways. Trivially, in the domain of reviews it would be desirable to model antonyms explicitly, *e.g.*, no restaurant review should be simultaneously labeled as having good and bad food. Other relationships between properties, such as hierarchical structures, could also be considered.

Finally, we believe that the core idea of using free-text as a source of training labels has broad applicability. For example, online blog entries are often tagged with short keyphrases. Our technique could be used to standardize these tags, and assign keyphrases to untagged blogs, thereby enabling more sophisticated content search and analysis.

References

- D. M. Blei, A. Y. Ng, M. I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.
- J. Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.
- I. Dagan, O. Glickman, B. Magnini. 2006. The PASCAL recognising textual entailment challenge. *Lecture Notes in Computer Science*, 3944:177–190.
- J. R. Finkel, T. Grenager, C. Manning. 2005. Incorporating non-local information into information extrac-

- tion systems by Gibbs sampling. In *Proceedings of the ACL*, 363–370.
- A. Gelman, J. B. Carlin, H. S. Stern, D. B. Rubin. 2004. *Bayesian Data Analysis*. Texts in Statistical Science. Chapman & Hall/CRC, 2nd edition.
- S. Goldwater, T. L. Griffiths, M. Johnson. 2006. Contextual dependencies in unsupervised word segmentation. In *Proceedings of ACL*.
- M. Hu, B. Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of SIGKDD*, 168–177.
- S.-M. Kim, E. Hovy. 2006. Automatic identification of pro and con reasons in online reviews. In *Proceedings of the COLING/ACL*, 483–490.
- D. Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th ICML*, 296–304.
- A.-M. Popescu, B. Nguyen, O. Etzioni. 2005. OPINE: Extracting product features and opinions from reviews. In *Proceedings of HLT/EMNLP*.
- W. M. Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.
- K. Toutanova, M. Johnson. 2007. A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In *Advances in Neural Information Processing Systems 20*.