Learning Document-Level Semantic Properties from Free-text Annotations

S.R.K. Branavan Harr Chen Jacob Eisenstein Regina Barzilay
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
{branavan, harr, jacobe, regina}@csail.mit.edu

Abstract

This paper demonstrates a new method for leveraging free-text annotations to infer semantic properties of documents. Free-text annotations are becoming increasingly abundant, due to the recent dramatic growth in semi-structured, user-generated online content. An example of such content is product reviews, which are often annotated by their authors with pros/cons keyphrases such as "a real bargain" or "good value." To exploit such noisy annotations, we simultaneously find a hidden paraphrase structure of the keyphrases, a model of the document texts, and the underlying semantic properties that link the two. This allows us to predict properties of unannotated documents. Our approach is implemented as a hierarchical Bayesian model with joint inference, which increases the robustness of the keyphrase clustering and encourages the document model 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 this type 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.

With the rapid increase of online content created by end users, noisy free-text annotations have become widely available (Vickery and Wunsch-Vincent, 2007; Sterling, 2005). For example, consider reviews of consumer products and services. 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. Having such a system would facilitate structured access and summarization of this data. However, novice-generated keyphrase annotations are incomplete descriptions of their corresponding review texts. Furthermore, they lack consistency: the same underlying property may be expressed in many ways, *e.g.*, "healthy" and "great nutritional value" (see Figure 1). 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.

pros/cons: *great nutritional value* ... combines it all: an amazing product, quick and friendly service, cleanliness, great nutrition ...
pros/cons: *a bit pricey, healthy* ... 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.

This paper presents a model that addresses both problems simultaneously. We assume that both the document text and the selection of keyphrases are governed by the underlying hidden properties of the document. Each property indexes a language model, thus allowing documents that incorporate the same

property to share similar features. In addition, each 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 lexical properties, and a hidden topic model is applied to the document text. Crucially, the keyphrase clusters and document 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 model is broadly applicable to many scenarios where documents are annotated in a noisy manner. In this work, we apply our method to a collection of reviews in two categories: restaurants and cell phones. The training data consists of review text and the associated pros/cons lists. We then evaluate the ability of our model to predict review properties when the pros/cons 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. However, our approach is distinguished in two ways: first, we are able to predict keyphrases beyond those that appear verbatim in the text. Second, our approach learns the relationships between 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 (LDA) (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.

Recent work has examined coupling topic models with explicit supervision (Blei and McAuliffe, 2007; Titov and McDonald, 2008). However, such approaches assume that the documents are labeled within a predefined annotation structure, *e.g.*, the properties of food, ambiance, and service for restaurants. In contrast, we address free-text annotations created by end users, without known semantic properties. Rather than requiring a predefined annotation structure, our model infers one from the data.

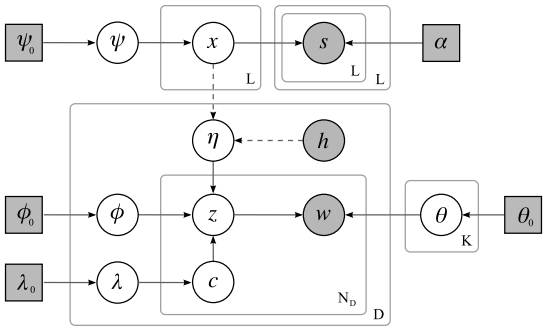
3 Problem Formulation

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 unseen 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 the properties supported by a previously unseen document. We can then use this property list to generate an appropriate set of keyphrases.

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 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 text.

We train the model on documents annotated with keyphrases. During training, we learn a hidden topic model from the text; each topic is also asso-



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 – document topic model
z – word topic assignment
θ – language models of each topic
w – document words

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

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

ciated 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.

4.1 Keyphrase Clustering

One of our goals is to cluster the keyphrases, such that each cluster corresponds to a well-defined property. 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, the plate diagram of our model.[1] Modeling the similarity matrix rather than the surface forms allows arbitrary comparisons between keyphrases, *e.g.*, permitting the use of both lexical and distributional information. The lexical comparison is based on the cosine similarity between the keyphrase words. The distributional similarity is quantified in terms of the co-occurrence of keyphrases across review texts. Our model is inherently capable of using any arbitrary source 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. To tie these topics to the keyphrases, we deterministically construct a document-specific topic distribu-

[1] We assume that similarity scores are conditionally independent given the keyphrase clustering, though the scores are in fact related. Such simplifying assumptions have been previously used with success in NLP (e.g., Toutanova and Johnson, 2007), though a more theoretically sound treatment of the similarity matrix is an area for future research.

tion 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 a small smoothing probability to other topics.

As noted above, properties may be expressed in the text even when no related keyphrase appears. For this reason, we also construct a document-specific topic distribution ϕ. The auxiliary variable c indicates whether a given word's topic is drawn from the set of keyphrase clusters, or from this topic distribution.

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 $\alpha_\neq$ otherwise. The parameters $\alpha_=$ linearly bias $s_{\ell,\ell'}$ towards one (Beta$(\alpha_=) \equiv$ Beta$(2,1)$), and the parameters $\alpha_\neq$ linearly bias $s_{\ell,\ell'}$ towards zero (Beta$(\alpha_\neq) \equiv$ Beta$(1,2)$).

Next, the words in each of the D documents are generated. Document d has N_d words; $z_{d,n}$ is the topic for word $w_{d,n}$. These latent topics are drawn either from the set of clusters represented by the document's keyphrases, or from the document's topic model ϕ_d. We deterministically construct a document-specific keyphrase topic model η_d, based on the keyphrase cluster assignments $\mathbf{x}$ and the observed keyphrases h_d. The multinomial η_d assigns equal probability to each topic that is represented by a phrase in h_d, and a small 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 document topic multinomial ϕ_d 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 document topic model ϕ_d. $c_{d,n}$ 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 $z_{d,n} \sim \phi_d$ 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, but sampling-based techniques can be used to estimate the posterior. We employ Gibbs sampling, previously used in NLP by Finkel et al. (2005) and Goldwater et al. (2006), among others. This technique repeatedly samples from the conditional distributions of each hidden variable, eventually converging on a Markov chain whose stationary distribution is the posterior distribution of the hidden variables in the model (Gelman et al., 2004). 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 $p(\psi \mid \ldots)$ to mean the probability conditioned on all the other variables.

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

where $\psi'_i = \psi_0 + \text{count}(x_\ell = i)$. 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 each keyphrase assignment x_ℓ from the others, given ψ.

ϕ_d and θ_k are resampled in a similar manner:

$$\begin{aligned} p(\phi_d \mid \ldots) &\propto \text{Dir}(\phi_d; \phi'_d), \\ p(\theta_k \mid \ldots) &\propto \text{Dir}(\theta_k; \theta'_k), \end{aligned}$$

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

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

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

$$
p(\lambda \mid \ldots) \propto \text{Beta}(\lambda; \lambda'),
$$

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

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 $\alpha_\neq$ 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 keyphrase topic distribution η_d, document topic distribution ϕ_d, words $\mathbf{w}$, and auxiliary variables $\mathbf{c}$:

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

As with x_ℓ, 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 each auxiliary variable $c_{d,n}$, which indicates whether the hidden topic $z_{d,n}$ is drawn from η_d or ϕ_d. The conditional probability for $c_{d,n}$ depends on its prior λ and the hidden topic assignments $z_{d,n}$:

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

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 Epinions[2] website. Users of this website evaluate products by providing both a textual description of their opinion, as well as concise lists of keyphrases (pros 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 mentioned in the text from the list of keyphrases. To obtain a complete gold

[2] `http://www.epinions.com/`

	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.

standard, we hand-annotated a subset of the reviews from the restaurant category. The annotation effort focused on eight commonly mentioned properties, such as those underlying the keyphrases "pleasant atmosphere" and "attentive staff." Two raters annotated 160 reviews, 30 of which were annotated by both. Cohen's kappa, a measure of interrater agreement ranging from zero to one, was 0.78 for this subset, indicating high agreement (Cohen, 1960).

Each review was annotated with 2.56 properties on average. Each manually-annotated property corresponded to an average of 19.1 keyphrases in the restaurant data, and 6.7 keyphrases in the cell phone data. This supports our intuition that a single semantic property may be expressed using a variety of different keyphrases.

Training 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 the restaurant data — where the gold standard identified eight semantic properties — we set K to 20, allowing the model to account for keyphrases not included in the eight most common properties. For the cell phones category, we set K to 30.

To improve the model's convergence rate, we perform two initialization steps for the Gibbs sampler. First, sampling is done only on the keyphrase clustering component of the model, ignoring document text. Second, we fix this clustering and sample the remaining model parameters. These two steps are run for 5,000 iterations each. The full joint model is then sampled for 100,000 iterations. Inspection of the parameter estimates confirms model convergence. On a 2GHz dual-core desktop machine, a multi-threaded C++ implementation of model training takes about two hours for each dataset.

Inference 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 single-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 threshold for this proportion is set for each property via the development set.

Evaluation 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, the model first predicts the properties supported by a given review. We then test whether the original authors' keyphrases are contained in the clusters associated with these properties.

As noted above, the authors' keyphrases are often incomplete. 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 they are supported by the model based on the document text.

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 (De Groot and Schervish, 2001).

Baselines To the best of our knowledge, this task not been previously 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. Because of this narrow requirement, precision should be high whereas recall will be low.

	Restaurants gold standard annotation			Restaurants free-text annotation			Cell Phones free-text annotation		
	Recall	Prec.	F-score	Recall	Prec.	F-score	Recall	Prec.	F-score
Random	0.500	0.300	$*$ 0.375	0.500	0.500	$*$ 0.500	0.500	0.489	$*$ 0.494
Phrase in text	0.048	0.500	$*$ 0.087	0.078	0.909	$*$ 0.144	0.171	0.529	$*$ 0.259
Cluster in text	0.223	0.534	0.314	0.517	0.640	$*$ 0.572	0.829	0.547	0.659
Phrase classifier	0.028	0.636	$*$ 0.053	0.068	0.963	$*$ 0.126	0.029	0.600	$*$ 0.055
Cluster classifier	0.113	0.622	$\diamond$ 0.192	0.255	0.907	$*$ 0.398	0.210	0.759	0.328
Our model	0.625	0.416	**0.500**	0.901	0.652	**0.757**	0.886	0.585	**0.705**
Our model + gold clusters	0.582	0.398	0.472	0.795	0.627	$*$ 0.701	0.886	0.520	$\diamond$ 0.655

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. Results for our model using the fixed, manually-created gold clusterings are also shown. The methods against which our model has significantly better results on the sign test are indicated with a $*$ for $p <= 0.05$, and $\diamond$ for $p <= 0.1$.

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: Discriminative classifiers are trained for each keyphrase. Positive examples are documents that are labeled 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: Discriminative classifiers are trained for each cluster of keyphrases, using our model's clustering. Positive examples are documents that are labeled 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.

Phrase classifier and *cluster classifier* employ maximum entropy classifiers, trained on the same features as our model, *i.e.*, word counts. 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, which should improve recall. We used Zhang Le's MaxEnt toolkit[3] to build these classifiers.

[3] `http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html`

7 Results

Comparative performance Table 2 presents the results of the 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 standard annotations shows that the random baseline outperforms all of the other baselines. We observe similar disappointing results for the non-random baselines against the free-text annotations. The precision and recall characteristics of the baselines 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 worse, with an F-score of 0.390 in the restaurant category and 0.171 in the cell phone category.

Impact of paraphrasing As previously observed in entailment research (Dagan et al., 2006), paraphrasing information contributes greatly to improved performance on semantic inference. This is

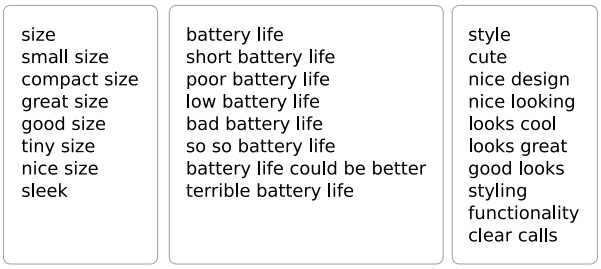


Figure 4: Sample keyphrase clusters that our model infers in the cell phone category.

confirmed by the dramatic difference in results between the *cluster in text* and *phrase in text* baselines. Therefore it is important to quantify the quality of automatically computed paraphrases, such as those illustrated in Figure 4.

	Restaurants	Cell Phones
Keyphrase similarity only	0.931	0.759
Joint training	**0.966**	**0.876**

Table 3: Rand Index scores of our model's clusters, using only keyphrase similarity vs. using keyphrases and text jointly. Comparison of cluster quality is against the gold standard.

One way to assess clustering quality is to compare it against a "gold standard" clustering, as constructed in Section 6. 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. Table 3 shows the Rand Indices for our model's clustering, as well as the clustering obtained by using only keyphrase similarity. These scores confirm that joint inference produces better clusters than using only keyphrases.

Another way of assessing cluster quality is to consider the impact of using the gold standard clustering instead of our model's clustering. As shown in the last two lines of Table 2, using the gold clustering yields results worse than using the model 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 hierarchical Bayesian model 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 system successfully extracts semantic properties of unannotated restaurant and cell phone reviews, empirically validating our approach.

Our present model makes strong assumptions about the independence of similarity scores. We believe this could be avoided by modeling the generation of the entire similarity matrix jointly. We have also assumed that the properties themselves are unstructured, but they are in fact related in interesting ways. For example, it would be desirable to model antonyms explicitly, *e.g.*, no restaurant review should be simultaneously labeled as having good and bad food. The correlated topic model (Blei and Lafferty, 2006) is one way to account for relationships between hidden topics; more structured representations, such as hierarchies, may also be considered.

Finally, the core idea of using free-text as a source of training labels has wide applicability, and has the potential to enable sophisticated content search and analysis. 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. The notion of free-text annotations is also very broad — we are currently exploring the applicability of this model to Wikipedia articles, using section titles as keyphrases, to build standard article schemas.

Acknowledgments

The authors acknowledge the support of the NSF, Quanta Computer, the U.S. Office of Naval Research, and DARPA. Thanks to Michael Collins, Dina Katabi, Kristian Kersting, Terry Koo, Brian Milch, Tahira Naseem, Dan Roy, Benjamin Snyder, Luke Zettlemoyer, and the anonymous reviewers for helpful comments and suggestions. Any opinions, findings, and conclusions or recommendations expressed above are those of the authors and do not necessarily reflect the views of the NSF.

References

David M. Blei and John D. Lafferty. 2006. Correlated topic models. In *Advances in NIPS*, pages 147–154.

David M. Blei and Jon McAuliffe. 2007. Supervised topic models. In *Advances in NIPS*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. *Lecture Notes in Computer Science*, 3944:177–190.

Morris H. De Groot and Mark J. Schervish. 2001. *Probability and Statistics*. Addison Wesley.

Jenny R. Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the ACL*, pages 363–370.

Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. 2004. *Bayesian Data Analysis*. Texts in Statistical Science. Chapman & Hall/CRC, 2nd edition.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2006. Contextual dependencies in unsupervised word segmentation. In *Proceedings of ACL*, pages 673–680.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of SIGKDD*, pages 168–177.

Soo-Min Kim and Eduard Hovy. 2006. Automatic identification of pro and con reasons in online reviews. In *Proceedings of the COLING/ACL*, pages 483–490.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of ICML*, pages 296–304.

Ana-Maria Popescu, Bao Nguyen, and Oren Etzioni. 2005. OPINE: Extracting product features and opinions from reviews. In *Proceedings of HLT/EMNLP*, pages 339–346.

William M. Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, December.

Bruce Sterling. 2005. Order out of chaos: What is the best way to tag, bag, and sort data? Give it to the unorganized masses. `http://www.wired.com/wired/archive/13.04/view.html?pg=4`. Accessed April 21, 2008.

Ivan Titov and Ryan McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of the ACL*.

Kristina Toutanova and Mark Johnson. 2007. A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In *Advances in NIPS*.

Graham Vickery and Sacha Wunsch-Vincent. 2007. *Participative Web and User-Created Content: Web 2.0, Wikis and Social Networking*. OECD Publishing.