

## Fine-Grained Class Label Markup of Search Queries

**Joseph Reisinger\***

Department of Computer Sciences  
The University of Texas at Austin  
Austin, Texas 78712  
joeraii@cs.utexas.edu

**Marius Paşca**

Google Inc.  
1600 Amphitheatre Parkway  
Mountain View, California 94043  
mars@google.com

### Abstract

We develop a novel approach to the semantic analysis of short text segments and demonstrate its utility on a large corpus of Web search queries. Extracting meaning from short text segments is difficult as there is little semantic redundancy between terms; hence methods based on shallow semantic analysis may fail to accurately estimate meaning. Furthermore search queries lack explicit syntax often used to determine intent in question answering. In this paper we propose a hybrid model of semantic analysis combining explicit class-label extraction with a latent class PCFG. This *class-label correlation* (CLC) model admits a robust parallel approximation, allowing it to scale to large amounts of query data. We demonstrate its performance in terms of (1) its predicted label accuracy on polysemous queries and (2) its ability to accurately chunk queries into base constituents.

### 1 Introduction

Search queries are generally short and rarely contain much explicit syntax, making query understanding a purely semantic endeavor. Furthermore, as in noun-phrase understanding, shallow lexical semantics is often irrelevant or misleading; e.g., the query [*tropical breeze cleaners*] has little to do with island vacations, nor are desert birds relevant to [*1970 road runner*], which refers to a car model.

This paper introduces *class-label correlation* (CLC), a novel unsupervised approach to extract-

ing shallow semantic content that combines class-based semantic markup (e.g., *road runner* is a *car model*) with a latent variable model for capturing weakly compositional interactions between query constituents. Constituents are tagged with IsA class labels from a large, automatically extracted lexicon, using a probabilistic context free grammar (PCFG). Correlations between the resulting label→term distributions are captured using a set of latent production rules specified by a hierarchical Dirichlet Process (Teh et al., 2006) with latent data groupings.

Concretely, the IsA tags capture the inventory of potential meanings (e.g., *jaguar* can be labeled as *european car* or *large cat*) and relevant constituent spans, while the latent variable model performs sense and theme disambiguation (e.g., [*jaguar habitat*] would lend evidence for the *large cat* label). In addition to broad sense disambiguation, CLC can distinguish closely related usages, e.g., the use of *dell* in [*dell motherboard replacement*] and [*dell stock price*].<sup>1</sup> Furthermore, by employing IsA class labeling as a preliminary step, CLC can account for common non-compositional phrases, such as *big apple* unlike systems relying purely on lexical semantics. Additional examples can be found later, in Figure 5.

In addition to improving query understanding, potential applications of CLC include: (1) relation extraction (Baeza-Yates and Tiberi, 2007), (2) query substitutions or broad matching (Jones et al., 2006), and (3) classifying other short textual fragments such as SMS messages or tweets.

We implement a parallel inference procedure for

\*Contributions made during an internship at Google.

<sup>1</sup>Dell the *computer system* vs. Dell the *technology company*.

CLC and evaluate it on a sample of 500M search queries along two dimensions: (1) query constituent chunking precision (i.e., how accurate are the inferred spans breaks; cf., Bergsma and Wang (2007); Tan and Peng (2008)), and (2) class label assignment precision (i.e., given the query intent, how relevant are the inferred class labels), paying particular attention to cases where queries contain ambiguous constituents. CLC compares favorably to several simpler submodels, with gains in performance stemming from coarse-graining related class labels and increasing the number of clusters used to capture between-label correlations.

**(Paper organization):** Section 2 discusses relevant background, Section 3 introduces the CLC model, Section 4 describes the experimental setup employed, Section 5 details results, Section 6 introduces areas for future work and Section 7 concludes.

## 2 Background

Query understanding has been studied extensively in previous literature. Li (2010) defines the semantic structure of noun-phrase queries as *intent heads* (attributes) coupled with some number of *intent modifiers* (attribute values), e.g., the query [*alice in wonderland 2010 cast*] is comprised of an intent head *cast* and two intent modifiers *alice in wonderland* and *2010*. In this work we focus on semantic class markup of query constituents, but our approach could be easily extended to account for query structure as well.

Popescu et al. (2010) describe a similar class-label-based approach for query interpretation, explicitly modeling the importance of each label for a given entity. However, details of their implementation were not publicly available, as of publication of this paper.

For simplicity, we extract class labels using the seed-based approach proposed by Van Durme and Paşca (2008) (in particular Paşca (2010)) which generalizes Hearst (1992). Talukdar and Pereira (2010) use graph-based semi-supervised learning to acquire class-instance labels; Wang et al. (2009) introduce a similar CRF-based approach but only apply it to a small number of verticals (i.e., *Computing and Electronics* or *Clothing and Shoes*). Snow et al. (2006) describe a learning approach for automatically ac-

quiring patterns indicative of hypernym (IsA) relations. Semantic class label lexicons derived from any of these approaches can be used as input to CLC.

Several authors have studied query clustering in the context of information retrieval (e.g., Beeferman and Berger, 2000). Our approach is novel in this regard, as we cluster queries in order to capture correlations between span labels, rather than explicitly for query understanding.

Tratz and Hovy (2010) propose a taxonomy for classifying and interpreting noun-compounds, focusing specifically on the relationships holding between constituents. Our approach yields similar topical decompositions of noun-phrases in queries and is completely unsupervised.

Jones et al. (2006) propose an automatic method for *query substitution*, i.e., replacing a given query with another query with the similar meaning, overcoming issues with poor paraphrase coverage in tail queries. Correlations mined by our approach are readily useful for downstream query substitution.

Bergsma and Wang (2007) develop a supervised approach to query chunking using 500 hand-segmented queries from the AOL corpus. Tan and Peng (2008) develop a generative model of query segmentation that makes use of a language model and concepts derived from Wikipedia article titles. CLC differs fundamentally in that it learns concept label markup in addition to segmentation and uses in-domain concepts derived from queries themselves. This work also differs from both of these studies significantly in scope, training on 500M queries instead of just 500.

At the level of class-label markup, our model is related to Bayesian PCFGs (Liang et al., 2007; Johnson et al., 2007b), and is a particular realization of an *Adaptor Grammar* (Johnson et al., 2007a; Johnson, 2010).

Szpektor et al. (2008) introduce a model of *contextual preferences*, generalizing the notion of selectional preference (cf. Ritter et al., 2010) to arbitrary terms, allowing for context-sensitive inference. Our approach differs in its use of class-instance labels for generalizing terms, a necessary step for dealing with the lack of syntactic information in queries.

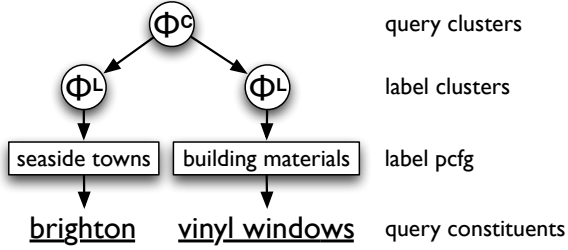


Figure 1: Overview of CLC markup generation for the query *[brighton vinyl windows]*. Arrows denote multinomial distributions.

### 3 Latent Class-Label Correlation

Input to CLC consists of raw search queries and a partial grammar mapping class labels to query spans (e.g., *building materials*→*vinyl windows*). CLC infers two additional latent productions types on top of these class labels: (1) a potentially infinite set of label clusters  $\phi_{l_k}^L$  coarse-graining the raw input label productions  $V$ , and (2) a finite set of query clusters  $\phi_{c_i}^C$  specifying distributions over label clusters; see Figure 1 for an overview.

Operationally, CLC is implemented as a Hierarchical Dirichlet Process (HDP; Teh et al., 2006) with latent groups coupled with a Probabilistic Context Free Grammar (PCFG) likelihood function (Figure 2). We motivate our use of an HDP latent class model instead of a full PCFG with binary productions by the fact that the space of possible binary rule combinations is prohibitively large (561K base labels; 314B binary rules). The next sections discuss the three main components of CLC: §3.1 the raw IsA class labels, §3.2 the PCFG likelihood, and §3.3 the HDP with latent groupings.

#### 3.1 IsA Label Extraction

IsA class labels (hypernyms)  $V$  are extracted from a large corpus of raw Web text using the method proposed by Van Durme and Paşca (2008) and extended by Paşca (2010). Manually specified patterns are used to extract a seed set of class labels and the resulting label lists are reranked using cluster purity measures. 561K labels for base noun phrases are collected. Table 1 shows an example set of class labels extracted for several common noun phrases. Similar repositories of IsA labels, extracted using other methods, are available for experimental pur-

class label→query span
recreational facilities→jacuzzi
rural areas→wales
destinations→wales
seaside towns→brighton
building materials→vinyl windows
consumer goods→european clothing

Table 1: Example production rules collected using the semi-supervised approach of Van Durme and Paşca (2008).

poses (Talukdar and Pereira, 2010). In addition to extracted rules, the CLC grammar is augmented with a set of *null rules*, one per unigram, ensuring that every query has a valid parse.

#### 3.2 Class-Label PCFG

In addition to the observed class-label production rules, CLC incorporates two sets of latent production rules coupled via an HDP (Figure 1). Class label→query span productions extracted from raw text are clustered into a set of latent *label production clusters*  $\mathcal{L} = \{l_1, \dots, l_\infty\}$ . Each label production cluster  $l_k$  defines a multinomial distribution over class labels  $V$  parametrized by  $\phi_{l_k}^L$ . Conceptually,  $\phi_{l_k}^L$  captures a set of class labels with similar productions that are found in similar queries, for example the class labels *states*, *northeast states*, *u.s. states*, *state areas*, *eastern states*, and *certain states* might be included in the same coarse-grained cluster due to similarities in their productions.

Each query  $q \in \mathcal{Q}$  is assigned to a latent *query cluster*  $c_q \in \mathcal{C}\{c_1, \dots, c_\infty\}$ , which defines a distribution over label production clusters  $\mathcal{L}$ , denoted  $\phi_{c_q}^C$ . Query clusters capture broad correlations between label production clusters and are necessary for performing sense disambiguation and capturing selectional preference. Query clusters and label production clusters are linked using a single HDP, allowing the number of label clusters to vary over the course of Gibbs sampling, based on the variance of the underlying data (Section 3.3). Viewed as a grammar, CLC only contains unary rules mapping labels to query spans; production correlations are captured directly by the query cluster, unlike in HDP-PCFG (Liang et al., 2007), as branching parses over the en-

		Indices	Cardinality
HDP base measure	$\beta \sim \text{GEM}(\gamma)$	-	$ \mathcal{L}  \rightarrow \infty$
Query cluster	$\phi_i^C \sim \text{DP}(\alpha^C, \beta)$	$i \in  \mathcal{C} $	$ \mathcal{L}  \rightarrow \infty$
Label cluster	$\phi_k^L \sim \text{Dirichlet}(\alpha^L)$	$k \in  \mathcal{L} $	$ V $
Query cluster ind	$\pi_q \sim \text{Dirichlet}(\xi)$ $c_q \sim \pi_q$	$q \in  \mathcal{Q} $ $q \in  \mathcal{Q} $	$ \mathcal{C} $ 1
Label cluster ind	$z_{q,t} \sim \phi_{c_q}^C$	$\mathbf{t} \in q, q \in  \mathcal{Q} $	1
Label ind	$l_{q,t} \sim \phi_{z_{q,t}}^L$	$\mathbf{t} \in q, q \in  \mathcal{Q} $	1

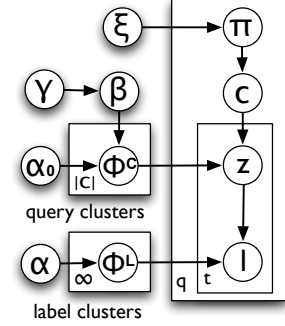


Figure 2: Generative process and graphical model for CLC. The top section of the model is the standard HDP prior; the middle section is the additional machinery necessary for modeling latent groupings and the bottom section contains the indicators for the latent class model. PCFG likelihood is not shown.

tire label sparse are intractably large.

Given a query  $q$ , a query cluster assignment  $c_q$  and a set of label production clusters  $\mathcal{L}$ , we define a parse of  $q$  to be a sequence of productions  $\mathbf{t}_q$  forming a parse tree consuming all the tokens in  $q$ . As with Bayesian PCFGs (Johnson, 2010), the probability of a tree  $\mathbf{t}_q$  is the product of the probabilities of the production rules used to construct it

$$P(\mathbf{t}_q | \phi^L, \phi^C, c_q) = \prod_{r \in R_q} P(r | \phi_{l_r}^L) P(l_r | \phi_{c_q}^C)$$

where  $R_q$  is the set of production rules used to derive  $\mathbf{t}_q$ ,  $P(r | \phi_{l_r}^L)$  is the probability of  $r$  given its label cluster assignment  $l_r$ , and  $P(l_r | \phi_{c_q}^C)$  is the probability of label cluster  $l_r$  in query cluster  $c$ .

The probability of a query  $q$  is the sum of the probabilities of the parse trees that can generate it,

$$P(q | \phi^L, \phi^C, c_q) = \sum_{\{\mathbf{t} | y(\mathbf{t}) = q\}} P(\mathbf{t} | \phi^L, \phi^C, c_q)$$

where  $\{\mathbf{t} | y(\mathbf{t}) = q\}$  is the set of trees with  $q$  as their yield (i.e., generate the string of tokens in  $q$ ).

### 3.3 Hierarchical Dirichlet Process with Latent Groups

We complete the Bayesian generative specification of CLC with an HDP prior linking  $\phi^C$  and  $\phi^L$ . The HDP is a Bayesian generative model of shared structure for grouped data (Teh et al., 2006). A set of base clusters  $\beta \sim \text{GEM}(\gamma)$  is drawn from a Dirichlet Process with base measure  $\gamma$  using the stick-breaking construction, and clusters for each group  $k$ ,

- $\gamma$  – HDP-LG base-measure smoother; higher values lead to more uniform mass over label clusters.
- $\alpha^C$  – Query cluster smoothing; higher values lead to more uniform mass over label clusters.
- $\alpha^L$  – Label cluster smoothing; higher values lead to more label diversity within clusters.
- $\xi$  – Query cluster assignment smoothing; higher values lead to more uniform assignment.

Table 2: CLC-HDP-LG hyperparameters.

$\phi_k^C \sim \text{DP}(\beta)$ , are drawn from a separate Dirichlet Process with base measure  $\beta$ , defined over the space of label clusters. Data in each group  $k$  are conditionally independent given  $\beta$ . Intuitively,  $\beta$  defines a common “menu” of label clusters, and each query cluster  $\phi_k^C$  defines a separate distribution over the label clusters.

In order to account for variable query-cluster assignment, we extend the HDP model with *latent groupings*  $\pi_q \sim \text{Dir}(\xi)$  for each query. The resulting *Hierarchical Dirichlet Process with Latent Groups* (HDP-LG) can be used to define a set of query clusters over a set of (potentially infinite) base label clusters (Figure 2). Each query cluster  $\phi^C$  (latent group) assigns weight to different subsets of the available label clusters  $\phi^L$ , capturing correlations between them at the query level. Each query  $q$  maintains a distribution over query clusters  $\pi_q$ , capturing its affinity for each latent group. The full generative specification of CLC is shown in Figure 2; hyperparameters are shown in Table 2.

In addition to the full joint CLC model, we evalu-

ate several simpler models:

1. CLC-BASE – no query clusters, one label per label cluster.
2. CLC-DPMM – no query clusters,  $\text{DPMM}(\alpha^C)$  distribution over labels.
3. CLC-HDP-LG – full HDP-LG model with  $|C|$  query clusters over a potentially infinite number of query clusters.

as well as various hyperparameter settings.

### 3.4 Parallel Approximate Gibbs Sampler

We perform inference in CLC via Gibbs sampling, leveraging Multinomial-Dirichlet conjugacy to integrate out  $\pi$ ,  $\phi^C$  and  $\phi^L$  (Teh et al., 2006; Johnson et al., 2007b). The remaining indicator variables  $\mathbf{c}$ ,  $\mathbf{z}$  and  $\mathbf{l}$  are sampled iteratively, conditional on all other variable assignments. Although there are an exponential number of parse trees for a given query, this space can be sampled efficiently using dynamic programming (Finkel et al., 2006; Johnson et al., 2007b)

In order to apply CLC to Web-scale data, we implement an efficient parallel approximate Gibbs sampler in the MapReduce framework Dean and Ghemawat (2004). Each Gibbs iteration consists of a single MapReduce step for sampling, followed by an additional MapReduce step for computing marginal counts.<sup>2</sup> Relevant assignments  $\mathbf{c}$ ,  $\mathbf{z}$  and  $\mathbf{l}$  are stored locally with each query and are distributed across compute nodes. Each node is responsible only for resampling assignments for its local set of queries. Marginals are fetched opportunistically from a separate distributed hash server as they are needed by the sampler. Each Map step computes a single Gibbs step for 10% of the available data, using the marginals computed at the previous step. By resampling only 10% of the available data each iteration, we minimize the potentially negative effects of using the previous step’s marginal distribution.

## 4 Experimental Setup

### 4.1 Query Corpus

Our dataset consists of a sample of 450M English queries submitted by anonymous Web users to

<sup>2</sup>This approximation and architecture is similar to Smola and Narayanamurthy (2010).

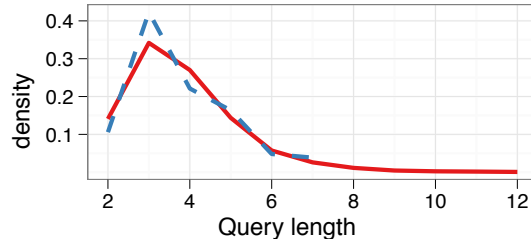


Figure 3: Distribution in the query corpus, broken down by query length (red/solid=all queries; blue/dashed=queries with ambiguous spans); most queries contain between 2-6 tokens.

Google. The queries have an average of 3.81 tokens per query (1.7B tokens). Single token queries are removed as the model is incapable of using context to disambiguate their meaning. Figure 3 shows the distribution of remaining queries. During training, we include 10 copies of each query (4.5B queries total), allowing an estimate of the Bayes average posterior from a single Gibbs sample.

### 4.2 Evaluations

Query markup is evaluated for phrase-chunking precision (Section 5.1) and label precision (Section 5.2) by human raters across two different samples: (1) an unbiased sample from the original corpus, and (2) a biased sample of queries containing ambiguous spans.

Two raters scored a total of 10K labels from 800 spans across 300 queries. Span labels were marked as *incorrect* (0.0), *badspan* (0.0), *ambiguous* (0.5), or *correct* (1.0), with numeric scores for label precision as indicated. Chunking precision is measured as the percentage of labels not marked as *badspan*.

We report two sets of precision scores depending on how *null* labels are handled: *Strict* evaluation treats null-labeled spans as incorrect, while *Normal* evaluation removes null-labeled spans from the precision calculation. Normal evaluation was included since the simpler models (e.g., CLC-BASE) tend to produce a significantly higher number of *null* assignments.

Model evaluations were broken down into *maximum a posteriori* (MAP) and *Bayes average* estimates. MAP estimates are calculated as the single most likely label/cluster assignment across all query copies; all assignments in the sample are averaged

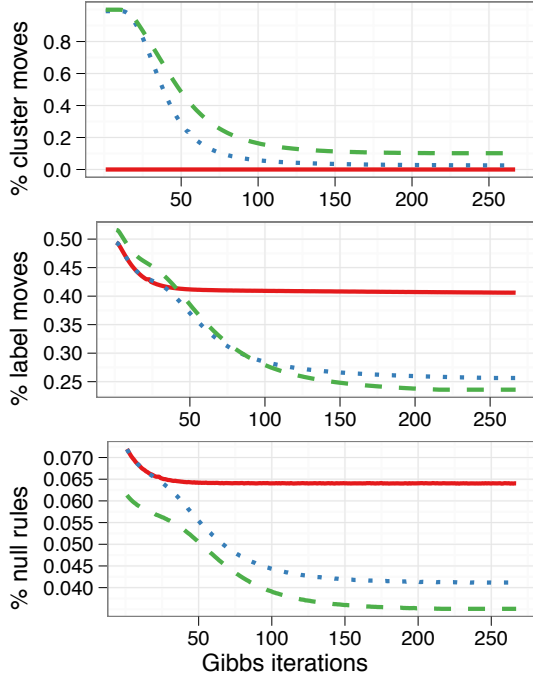


Figure 4: Convergence rates of CLC-BASE (red/solid), CLC-HDP-LG 100C,40L (green/dashed), CLC-HDP-LG 1000C,40L (blue/dotted) in terms of % of query cluster swaps, label cluster swaps and null rule assignments.

to obtain the Bayes average precision estimate.<sup>3</sup>

## 5 Results

A total of five variants of CLC were evaluated with different combinations of  $|C|$  and HDP prior concentration  $\alpha^C$  (controlling the effective number of label clusters). Referring to models in terms of their parametrizations is potentially confusing. Therefore, we will make use of the fact that models with  $\alpha^C = 1$  yielded roughly 40 label clusters on average, and models with  $\alpha^C = 0.1$  yielded roughly 200 label clusters, naming model variants simply by the number of query and label clusters: (1) CLC-BASE, (2) CLC-DPMM 1C-40L, (3) CLC-HDP-LG 100C-40L, (4) CLC-HDP-LG 1000C-40L, and (5) CLC-HDP-LG 1000C-200L. Figure 4 shows the model convergence for CLC-BASE, CLC-HDP-LG 100C-40L, and CLC-HDP-LG 1000C-40L.

<sup>3</sup>We calculate the Bayes average precision estimates at the top 10 (Bayes@10) and top 20 (Bayes@20) parse trees, weighted by probability.

### 5.1 Chunking Precision

Chunking precision scores for each model are shown in Table 3 (average % of labels not marked *badspan*). CLC-HDP-LG 1000C-40L has the highest precision across both MAP and Bayes estimates ( $\sim 93\%$  accuracy), followed by CLC-HDP-LG 1000C-200L ( $\sim 90\%$  accuracy) and CLC-DPMM 1C-40L ( $\sim 85\%$ ). CLC-BASE performed the worst by a significant margin ( $\sim 78\%$ ), indicating that label coarse-graining is more important than query clustering for chunking accuracy. No significant differences in label chunking accuracy were found between Bayes and MAP inference.

### 5.2 Predicting Span Labels

The full CLC-HDP-LG model variants obtain higher label precision than the simpler models, with CLC-HDP-LG 1000C-40L achieving the highest precision of the three ( $\sim 63\%$  accuracy). Increasing the number of label clusters too high, however, significantly reduces precision: CLC-HDP-LG 1000C-200L obtains only  $\sim 51\%$  accuracy. However, comparing to CLC-DPMM 1C-40L and CLC-BASE demonstrates that the addition of label clusters and query clusters both lead to gains in label precision. These relative rankings are robust across *strict* and *normal* evaluation regimes.

The breakdown over MAP and Bayes posterior estimation is less clear when considering label precision: the simpler models CLC-BASE and CLC-DPMM 1C-40L perform significantly worse than Bayes when using MAP estimation, while in CLC-HDP-LG the reverse holds.

There is little evidence for correlation between precision and query length (weak, not statistically significant negative correlation using Spearman’s  $\rho$ ). This result is interesting as the relative prevalence of natural language queries increases with query length, potentially degrading performance. However, we did find a strong positive correlation between precision and the number of labels productions applicable to a query, i.e., production rule fertility is a potential indicator of semantic quality.

Finally, the *histogram* column in Table 3 shows the distribution of rater responses for each model. In general, the more precise models tend to have a significantly lower proportion of missing spans

Model	Chunking Precision	Label Precision			Ambiguous Label Precision		Spearman's $\rho$	
		normal	strict	hist	normal	strict	q. len	# labels
Class-Label Correlation Base								
Bayes@10	78.7±1.1	37.7±1.2	35.8±1.2		35.4±2.0	33.2±1.9	-0.13	0.51 <sup>•</sup>
Bayes@20	78.7±1.1	37.7±1.2	35.8±1.2		35.4±2.0	33.2±1.9	-0.13	0.51 <sup>•</sup>
MAP	76.3±2.2	33.3±2.2	31.8±2.2		36.2±4.0	33.2±3.8	-0.13	0.52 <sup>•</sup>
Class-Label Correlation DPMM 1C 40L								
Bayes@10	84.9±0.4	46.6±0.6	44.3±0.5		36.0±1.1	33.7±1.0	-0.05	0.25
Bayes@20	84.8±0.4	47.4±0.5	45.2±0.5		37.8±1.0	35.5±1.0	-0.02	0.23
MAP	84.1±0.8	42.6±1.0	40.5±0.9		11.2±1.3	10.6±1.3	-0.03	0.12
Class-Label Correlation HDP-LG 100C 40L								
Bayes@10	83.8±0.4	55.6±0.5	51.0±0.5		<b>55.6±1.0</b>	<b>47.7±1.0</b>	0.03	0.44 <sup>•</sup>
Bayes@20	83.6±0.4	56.9±0.5	52.3±0.5		<b>57.4±1.0</b>	<b>49.8±0.9</b>	0.04	0.41 <sup>•</sup>
MAP	82.7±0.5	58.5±0.5	53.6±0.5		<b>60.4±1.1</b>	<b>51.5±1.0</b>	0.02	0.41 <sup>•</sup>
Class-Label Correlation HDP-LG 1000C 40L								
Bayes@10	<b>93.1±0.2</b>	<b>61.1±0.3</b>	<b>60.0±0.3</b>		43.2±0.9	40.2±0.9	-0.06	0.26 <sup>•</sup>
Bayes@20	<b>92.8±0.2</b>	<b>62.6±0.3</b>	<b>61.7±0.3</b>		44.9±0.8	42.2±0.8	-0.10	0.27 <sup>•</sup>
MAP	<b>92.7±0.2</b>	<b>63.7±0.3</b>	<b>62.7±0.3</b>		44.1±0.9	41.1±0.9	-0.12	0.28 <sup>•</sup>
Class-Label Correlation HDP-LG 1000C 200L								
Bayes@10	90.3±0.5	50.9±0.8	48.6±0.7		45.8±1.5	42.5±1.3	-0.10	0.13
Bayes@20	89.9±0.5	50.2±0.7	48.0±0.7		44.4±1.4	41.3±1.3	-0.08	0.11
MAP	90.0±0.6	51.0±0.8	48.9±0.8		49.2±1.5	46.0±1.4	-0.07	0.04

Table 3: Chunking and label precision across five models. Confidence intervals are standard error; sparklines show distribution of precision scores (left is zero, right is one). *Hist* shows the distribution of human rating response (log y scale): green/first is correct, blue/second is ambiguous, cyan/third is missing and red/fourth is incorrect. Spearman's  $\rho$  columns give label precision correlations with query length (weak negative correlation) and the number of applicable labels (weak to strong positive correlation); dots indicate significance.

(blue/second bar; due to null rule assignment) in addition to more correct (green/first) and fewer incorrect (red/fourth) spans.

### 5.3 High Polysemy Subset

We repeat the analysis of label precision on a subset of queries containing one of the manually-selected polysemous spans shown in Table 4. The CLC-HDP-LG -based models still significantly outperform the simpler models, but unlike in the broader setting, CLC-HDP-LG 100C-40L significantly outperforms CLC-HDP-LG 1000C-40L, indicating that lower query cluster granularity helps address polysemy (Table 3).

### 5.4 Error Analysis

Figure 5 gives examples of both high-precision and low-precision queries markups inferred by CLC-HDP-LG. In general, CLC performs well on queries with clear *intent head* / *intent modifier* structure (Li,

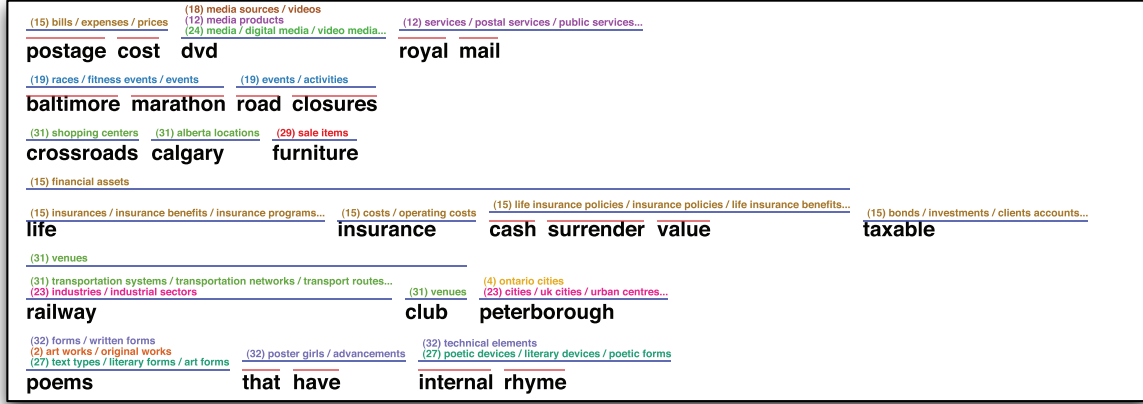
acapella, alamo, apple, atlas, bad, bank, batman, beloved, black forest, bravo, bush, canton, casino, champion, club, comet, concord, dallas, diamond, driver, english, ford, gamma, ion, lemon, manhattan, navy, pa, palm, port, put, resident evil, ronaldo, sacred heart, saturn, seven, solution, so-pranos, sparta, supra, texas, village, wolf, young

Table 4: Samples from a list of 90 manually selected ambiguous spans used to evaluate model performance under polysemy.

2010). More complex queries, such as [*never know until you try quotes*] or [*how old do you have to be a bartender in new york*] do not fit this model; however, expanding the set of extracted labels to also cover instances such as *never know until you try* would mitigate this problem, motivating the use of n-gram language models with semantic markup.

A large number of mistakes made by CLC are

Top 10%



Middle 20%

Bottom 20%

Figure 5: Examples of high- and low-precision query markups inferred by CLC-HDP-LG. Black text is the original query; lines indicate potential spans; small text shows potential labels colored and numbered by label cluster; small bar shows percentage of assignments to that label cluster.

due to named-entity categories with weak semantics such as rock bands or businesses (e.g., [*tropical breeze cleaners*], [*cosmic railroad band*] or [*sopranos cigars*]). When the named entity is common enough, it is detected by the rule set, but for the long tail of named entities this is not the case. One potential solution is to use a stronger notion of *selectional preference* and slot-filling, rather than just relying on correlation between labels.

Other examples of common errors include interpreting *weymouth* in [*weymouth train time table*] as a town in Massachusetts instead of a town in the UK (lack of domain knowledge), and using lower qual-

ity semantic labels (e.g., *neighboring countries* for *france*, or *great retailers* for *target*).

## 6 Discussion and Future Work

Adding both latent label clusters (DPMM) and latent query clusters (extending to HDP-LG) improve chunking and label precision over the baseline CLC-BASE system. The label clusters are important because they capture intra-group correlations between class labels, while the query clusters are important for capturing inter-group correlations. However, the algorithm is sensitive to the relative number of clusters in each case: Too many labels/label clusters rel-

ative to the number of query clusters make it difficult to learn correlations ( $O(n^2)$  query clusters are required to capture pairwise interactions). Too many query clusters, on the other hand, make the model intractable computationally. The HDP automates selecting the number of clusters, but still requires manual hyperparameter setting.

**(Future Work)** Many query slots have weak semantics and hence are misleading for CLC. For example [*pacific breeze cleaners*] or [*dale hartley subaru*] should be parsed such that the type of the leading slot is determined not by its direct content, but by its context; seeing *subaru* or *cleaners* after a noun-phrase slot is a strong indicator of its type (*dealership* or *shop name*). The current CLC model only couples these slots through their correlations in query clusters, not directly through relative position or context. Binary productions in the PCFG or a discriminative learning model would help address this.

Finally, we did not measure label coverage with respect to a human evaluation set; coverage is useful as it indicates whether our inferred semantics are biased with respect to human norms.

## 7 Conclusions

We introduced CLC, a set of latent variable PCFG models for semantic analysis of short textual segments. CLC captures semantic information in the form of interactions between clusters of automatically extracted class-labels, e.g., finding that place-names commonly co-occur with business-names. We applied CLC to a corpus containing 500M search queries, demonstrating its scalability and straightforward parallel implementation using frameworks like MapReduce or Hadoop. CLC was able to chunk queries into spans more accurately and infer more precise labels than several sub-models even across a highly ambiguous query subset. The key to obtaining these results was coarse-graining the input class-label set and using a latent variable model to capture interactions between coarse-grained labels.

## References

R. Baeza-Yates and A. Tiberi. 2007. Extracting semantic relations from query logs. In *Proceedings of the 13th ACM Conference on Knowledge Discovery and Data Mining (KDD-07)*, pages 76–85. San Jose, California.

D. Beeferman and A. Berger. 2000. Agglomerative clustering of a search engine query log. In *Proceedings of the 6th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD-00)*, pages 407–416.

S. Bergsma and Q. Wang. 2007. Learning noun phrase query segmentation. In *Proceedings of the 2007 Conference on Empirical Methods in Natural Language Processing (EMNLP-07)*, pages 819–826. Prague, Czech Republic.

J. Dean and S. Ghemawat. 2004. MapReduce: Simplified data processing on large clusters. In *Proceedings of the 6th Symposium on Operating Systems Design and Implementation (OSDI-04)*, pages 137–150. San Francisco, California.

J. Finkel, C. Manning, and A. Ng. 2006. Solving the problem of cascading errors: Approximate Bayesian inference for linguistic annotation pipelines. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP-06)*, pages 618–626. Sydney, Australia.

M. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pages 539–545. Nantes, France.

M. Johnson. 2010. PCFGs, topic models, adaptor grammars and learning topical collocations and the structure of proper names. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*, pages 1148–1157. Uppsala, Sweden.

M. Johnson, T. Griffiths, and S. Goldwater. 2007a. Adaptor grammars: a framework for specifying compositional nonparametric bayesian models. In *Advances in Neural Information Processing Systems 19*, pages 641–648. Vancouver, Canada.

M. Johnson, T. Griffiths, and S. Goldwater. 2007b. Bayesian inference for PCFGs via Markov Chain Monte Carlo. In *Proceedings of the 2007 Conference of the North American Association for Computational Linguistics (NAACL-HLT-07)*, pages 139–146. Rochester, New York.

R. Jones, B. Rey, O. Madani, and W. Greiner. 2006. Generating query substitutions. In *Proceedings of the 15th World Wide Web Conference (WWW-06)*, pages 387–396. Edinburgh, Scotland.

X. Li. 2010. Understanding the semantic structure of noun phrase queries. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*, pages 1337–1345. Uppsala, Sweden.

- P. Liang, S. Petrov, M. Jordan, and D. Klein. 2007. The infinite PCFG using hierarchical Dirichlet processes. In *Proceedings of the 2007 Conference on Empirical Methods in Natural Language Processing (EMNLP-07)*, pages 688–697. Prague, Czech Republic.
- M. Paşca. 2010. The role of queries in ranking labeled instances extracted from text. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING-10)*, pages 955–962. Beijing, China.
- A. Popescu, P. Pantel, and G. Mishne. 2010. Semantic lexicon adaptation for use in query interpretation. In *Proceedings of the 19th World Wide Web Conference (WWW-10)*, pages 1167–1168. Raleigh, North Carolina.
- A. Ritter, Mausam, and O. Etzioni. 2010. A latent Dirichlet allocation method for selectional preferences. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*, pages 424–434. Uppsala, Sweden.
- A. Smola and S. Narayanamurthy. 2010. An architecture for parallel topic models. In *Proceedings of the 36th Conference on Very Large Data Bases (VLDB-10)*, pages 703–710. Singapore.
- R. Snow, D. Jurafsky, and A. Ng. 2006. Semantic taxonomy induction from heterogeneous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, pages 801–808. Sydney, Australia.
- I. Szpektor, I. Dagan, R. Bar-Haim, and J. Goldberger. 2008. Contextual preferences. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL-08)*, pages 683–691. Columbus, Ohio.
- P. Talukdar and F. Pereira. 2010. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*, pages 1473–1481. Uppsala, Sweden.
- B. Tan and F. Peng. 2008. Unsupervised query segmentation using generative language models and Wikipedia. In *Proceedings of the 17th World Wide Web Conference (WWW-08)*, pages 347–356. Beijing, China.
- Y. Teh, M. Jordan, M. Beal, and D. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.
- S. Tratz and E. Hovy. 2010. A taxonomy, dataset, and classifier for automatic noun compound interpretation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*, pages 678–687. Uppsala, Sweden.
- B. Van Durme and M. Paşca. 2008. Finding cars, goddesses and enzymes: Parametrizable acquisition of labeled instances for open-domain information extraction. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI-08)*, pages 1243–1248. Chicago, Illinois.
- T. Wang, R. Hoffmann, X. Li, and J. Szymanski. 2009. Semi-supervised learning of semantic classes for query understanding: from the Web and for the Web. In *Proceedings of the 18th International Conference on Information and Knowledge Management (CIKM-09)*, pages 37–46. Hong Kong, China.