Topic Modeling of Freelance Job Postings to Monitor Web Service Abuse

Do-kyum Kim, Marti Motoyama, Geoffrey M. Voelker and Lawrence K. Saul

Department of Computer Science and Engineering
University of California, San Diego

ABSTRACT

Web services such as Google, Facebook, and Twitter are recurring victims of abuse, and their plight will only worsen as more attackers are drawn to their large user bases. Many attackers hire cheap, human labor to actualize their schemes, connecting with potential workers via crowdsourcing and freelancing sites such as Mechanical Turk and Freelancer.com. To identify solicitations for abuse jobs, these Web sites need ways to distinguish these tasks from ordinary jobs. In this paper, we show how to discover clusters of abuse tasks using latent Dirichlet allocation (LDA), an unsupervised method for topic modeling in large corpora of text. Applying LDA to hundreds of thousands of unlabeled job postings from Freelancer.com, we find that it discovers clusters of related abuse jobs and identifies the prevalent words that distinguish them. Finally, we use the clusters from LDA to profile the population of workers who bid on abuse jobs and the population of buyers who post their project descriptions.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; I.2.6 [**Artificial Intelligence**]: Learning—*Parameter learning*

General Terms

Algorithms, Experimentation, Security

Keywords

Latent Dirichlet allocation, crowdsourcing

1. INTRODUCTION

Many online Web services are free, generating revenue by serving as effective advertising channels and relying on users to provide interesting content for their sites. This open access allows companies to attract large numbers of users but also provides attackers with an opportunity to abuse their services. Web service abuse takes many forms, including creating fake accounts on Facebook to spam other users, launching deceptive advertising campaigns on Craigslist, and spamming comments sections on blogs as a form of blackhat search engine optimization (SEO). Web service providers counter this abuse by deploying new defenses (phone verification, different CAPTCHA types, etc.), but attackers respond with equal vigor, often utilizing brute force human labor to actualize their exploits. The problem is worsening as abusers use crowdsourcing sites to outsource abusive tasks to workers in low cost labor markets. Recent studies have estimated that 41% of jobs posted on Mechanical Turk [17] consisted of spam-related tasks and that 30% of jobs on Freelancer.com [22] involved service abuse.

In response, crowdsourcing sites have taken measures to identify and filter abuse jobs,[1] but the widespread prevalence of abuse job postings suggests that current defenses are ineffective. As one method for automating this process, previous work explored a supervised learning approach to identify and classify abuse jobs on Freelancer.com [22]. Although feasible as an initial study, this approach required significant manual effort to identify job categories and label job postings for training data. This level of manual effort would be difficult to scale to the demands of a continuous operational service for monitoring abuse.

In this paper, we explore an alternative method that drastically reduces the amount of manual labeling needed to discover clusters of abuse jobs from their free-form project descriptions. We use latent Dirichlet allocation (LDA) [4] to analyze more than 350,000 job postings on Freelancer.com, a large freelancing site with a substantial user population. LDA is a statistical topic model that clusters unlabeled documents based on the frequencies that different words appear in them. Viewing the job postings on Freelancer.com as unlabeled documents, we use LDA to discover clusters of abuse jobs and the keyword sets that identify them.

LDA is an *unsupervised* approach to topic modeling that works on unlabeled job postings. We compare the results from LDA to the results from supervised approaches that depend on an initial manual labeling of job postings. We find that LDA is not only able to cluster related jobs, but also to identify keywords (for each cluster) that provide insight into the targets and methods of abuse. We also use LDA to trace the evolution of demand for specific types of jobs over time. LDA does have limitations: in a few cases, LDA merges distinct job categories into the same cluster or splits a single job category into two clusters. To address these limitations, we also explore ways to identify mergeable topics. Overall, we conclude that LDA can significantly reduce the manual effort required to discover abuse jobs. It thus provides an operational tool for crowdsourcing sites to monitor and control the influx of abuse-related job postings.

2. BACKGROUND

Many crowdsourcing and freelancing sites exist today. The most well-known crowdsourcing site, Mechanical Turk, is mainly used

AISec'11, October 21, 2011, Chicago, Illinois, USA.
Copyright 2011 ACM 978-1-4503-1003-1/11/10 ...$10.00.

[1] In previous work, for instance, Freelancer.com invalidated job postings by one of the authors about Craigslist accounts.

for simple tasks; it is especially popular among researchers who need to collect and label large data sets of images and text. Unlike Mechanical Turk, freelancing sites such as Freelancer, Guru, and oDesk offer the ability to commission more complex jobs that involve Web site programming, mobile application development, etc. The dynamics of these sites are roughly the same: a buyer describes a job, after which workers vie to complete the task. In this paper, we investigate the content on Freelancer.com, a site advertising over a million job postings and two million freelance professionals [12]. We focus on Freelancer because it offers an API that allows us to gather comprehensive, historical data about its users and job listings (and also because we can compare results with previous work).

New users to Freelancer register on the site simply by providing a valid email address. To post a job, a buyer pays an initial \$5 fee, after which workers ("bidders") bid on the task, including short descriptions and prices in each of their bids. To complete the job, the buyer selects one worker based on a number of factors, including the prices, ratings, and backgrounds of bidding workers. At this point, Freelancer refunds the initial \$5 fee, but charges either \$3 or 3% of the total project budget (whichever is higher). Freelancer thus serves as the middleman, facilitating the exchange of money between the buyer and the selected worker. To sidestep Freelancer's cut of the payment, buyers and workers sometimes use Freelancer to rendezvous, then continue negotiations off the site.

Previous studies have relied on manual efforts to identify and label abuse jobs [17, 22]. In particular, these studies identified a number of pre-defined categories based on prior knowledge of Web service abuse, then labeled a small subset of jobs, indicating whether or not they belonged to one of these categories. The reliance on manual efforts has two problems. First, such studies are liable to overlook important categories of abuse jobs (i.e., ones not already known to exist) as they tend to focus on small numbers of categories (between six and 17). Second, such studies are limited by the amount of effort required to read and assign labels to individual job postings. To estimate the fraction of abuse jobs in their data sets, Ipeirotis et al. sample 100 postings, while Motoyama et al. manually label 2,000 jobs. The topic models that we explore in this paper address both these problems. They do not rely on prior knowledge to assign labels (only to interpret their results), and they scale well: we directly apply them to over 350,000 job postings.

In this paper we use the data set from [22], summarized in Table 1. The data set was obtained by programmatically crawling Freelancer.com using the site's own API. It contains over seven years of data on job postings and user profiles. While the site claims to have over two million users, the full data set contains only 800 thousand active users. Not all projects are viewable through the API as job postings are occasionally deleted by buyers (for jobs no longer needed) or Freelancer itself (due to abuse).

Activity	Count	
Projects	842,199	
Projects w/ Selected Workers	388,733	(46%)
Project Bids	12,656,978	
Active Users	815,709	
Buyers Only	179,908	(22.1%)
Workers Only	590,806	(72.4%)
Buyer & Workers	44,995	(5.5%)

Table 1: Summary of Freelancer activity between February 5, 2004 and April 6, 2011.

3. TOPIC MODELS

Topic modeling is an automatic, data-driven approach for analyzing and organizing large corpora of text. The most popular topic models—and the ones we consider in this paper—are trained from collections of *unlabeled* documents; that is to say, the documents have not already been manually categorized by human readers. The primary goal of topic modeling is to discover clusters of documents on similar subjects. These clusters are discovered by analyzing the frequencies of words that appear in different documents; when groups of words occur together frequently in some documents, but not in others, they are interpreted as keywords of distinct topics that appear in the corpus. Once trained from observed word counts and co-occurrences, a statistical topic model can be used to label each document in the corpus by its inferred topics. For each document in the corpus, the model can also be used to label individual words by the topics they most likely suggest.

For practitioners, topic models can be viewed as a black box that take as input a collection of unlabeled documents and return as output a distribution (or weighted list) of topics for each document. The main adjustable parameter of this black box is the total number of possible topics to be discovered in the corpus; in practice, this number is set by prior (domain) knowledge or some other validation criterion.

More formally, statistical topic models are a special class of probabilistic graphical models whose hidden variables are used to discover latent semantic structure in large corpora of text [2]. The latent semantic structure emerges naturally from the varied topics that appear in diverse collections of unlabeled documents. The "topics" in these models are represented as distributions over words, and the documents are viewed as having been generated from particular mixtures of topics.

The results of topic modeling are perhaps best illustrated by example. In an earlier study, [20] used these models to analyze the email messages exchanged between two researchers. In one of the discovered topics, the most highly weighted words were PROPOSAL, DATA, BUDGET, and NSF; the authors interpreted this grouping of words as evidence of multiple messages on the subject of grant proposals. Likewise, in another of the topics, the most highly weighted words were TODAY, TOMORROW, TIME, and MEETING; they interpreted this grouping of words as evidence of multiple messages on meeting scheduling.

Early approaches to topic modeling include latent semantic indexing (LSI) [8] and probabilistic LSI (pLSI) [15]. Later, [4] introduced latent Dirichlet allocation (LDA), the model we consider in this paper, which can be viewed as the first properly Bayesian model for topic modeling. The introduction of LDA generated a flood of both algorithmic and applied work on topic modeling. LDA and its extensions have been applied to collections of scientific papers [5, 13], news articles [4], Web pages [5], and email messages [20]. They have also been applied to corpora of music [16], images [10] and video [23]. In this paper, we apply LDA to our collection of job postings from the Freelancer data set. Before doing so, however, we briefly review the fundamentals of this model.

3.1 Latent Dirichlet Allocation

In LDA, we model each document as a collection of words drawn from a dictionary (or vocabulary) of fixed size V. We encode the words in this dictionary as unary V-dimensional vectors in which exactly one entry is equal to one and all others are equal to zero; specifically, the vth word in the dictionary is encoded as the vector $w \in \{0,1\}^V$ in which $w^v = 1$ and $w^u = 0$ for $u \neq v$. A document is simply a sequence of N words, which we represent by $\mathbf{w} = (w_1, w_2, \ldots, w_N)$ where w_n denotes the nth word in the document. Finally, a corpus is a set of D documents, which we represent by $M = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_D\}$.

For each document:

1. Draw topic proportions $\theta_d \sim$ Dirichlet(α), a K-dimensional vector.
2. For each word,
   (a) Draw a topic assignment $z_{d,n} \sim$ Multinomial(θ_d), $z_{d,n} \in \{1, 2, \dots, K\}$
   (b) Draw a word $w_{d,n} \sim$ Multinomial($\beta_{z_{d,n}}$), $w_{d,n} \in \{(1, 0, \dots), (0, 1, \dots), \dots, (\dots, 0, 1)\}$

Figure 1: The generative process for LDA.

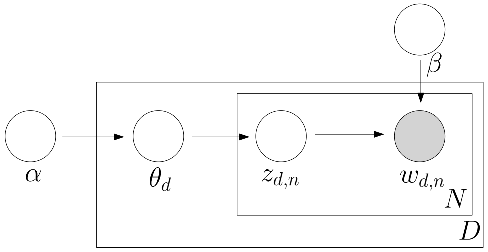


Figure 2: Representation of LDA as a graphical model: the nodes denote random variables, while the edges indicate conditional dependencies. The shaded nodes are observed variables (words); the unshaded nodes are hidden variables (topics). The outer rectangles, or "plates", indicate repeated samples.

LDA is a probabilistic generative model that hypothesizes a particular process by which documents in a large corpus are generated [4]. Some of the variables in this process are observed (such as the words $\mathbf{w}$ that appear in each document), while others are hidden (such as the topics, which must be inferred). Let K denote the number of topics that appear in the corpus. LDA has two model parameters: a K-dimensional vector α whose elements indicate the proportion of topics across the corpus as a whole, and a $K \times V$ matrix β whose elements indicate the probabilities that different topics give rise to different words. LDA imagines that (i) each document has a set of topic proportions θ_d, and that (ii) corresponding to each word $w_{d,n}$ in the dth document is a topic assignment $z_{d,n}$. Thus in addition to the N observed words $\mathbf{w}_d$ in each document, there are hidden topic assignments $\mathbf{z}_d = (z_{d,1}, z_{d,2}, \dots, z_{d,N})$. Figure 1 describes the overall generative process for LDA, while Figure 2 shows its representation as a graphical model.

As its name implies, LDA uses the Dirichlet distribution to allocate topic proportions for each document. A K-dimensional Dirichlet distribution is a probability density function over the $(K-1)$-dimensional simplex; thus, a sample from the distribution, θ, is a K-dimensional vector with $\theta_i \geq 0$ for all i and $\sum_{i=1}^{K} \theta_i = 1$. The Dirichlet distribution has the parametric form:

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1}, \tag{1}$$

where the parameter α is a positive K-dimensional vector, and $\Gamma(x)$ is the Gamma function. The expectation and variance of θ_i are given by:

$$\mathrm{E}[\theta_i|\alpha] = \frac{\alpha_i}{\sum_j \alpha_j}, \tag{2}$$

$$\mathrm{Var}[\theta_i|\alpha] = \frac{\mathrm{E}[\theta_i](1 - \mathrm{E}[\theta_i])}{1 + \sum_j \alpha_j}. \tag{3}$$

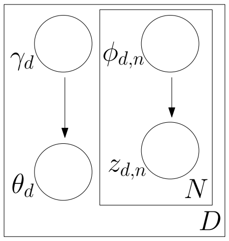


Figure 3: Graphical model for the variational approximation in LDA. The variational parameters are the Dirichlet parameter γ and the multinomial parameters ϕ.

A *symmetric* Dirichlet distribution uses the same value for each component of the parameter vector α. The Dirichlet distribution is a member of the exponential family; it is also the conjugate prior distribution for the multinomial distribution.

3.2 Variational Inference and Learning

The utility of LDA hinges on the ability to make inferences from the model. In particular, we must be able to ascertain how the topics in a document are revealed by the particular words that it contains. For the dth unlabeled document, this inference is performed by computing statistics of the posterior distribution over hidden topics given observed words:

$$p(\theta_d, z_d|\mathbf{w}_d, \alpha, \beta) = \frac{p(\theta_d, z_d, \mathbf{w}_d|\alpha, \beta)}{p(\mathbf{w}_d|\alpha, \beta)}. \tag{4}$$

The most important statistics from this distribution are the expected topic proportions $\mathrm{E}[\theta_d|\mathbf{w}_d, \alpha, \beta]$ and expected topic assignments $\mathrm{E}[z_d|\mathbf{w}_d, \alpha, \beta]$. However, as noted in [4], it is intractable to compute these statistics of the posterior distribution, or even to compute the denominator in eq. (4). In light of this intractability, it is necessary to adopt approximate methods for inference. There are two popular approaches for approximate inference in LDA: variational methods [4] and Gibbs sampling [13]. In this paper, we use variational methods, a brief review of which follows.

Variational methods are based on a simple idea: they approximate the intractable distribution in eq. (4) by a tractable one whose statistics are easy to compute [18]. The tractable distribution is chosen from a parameterized family of distributions, and within this family, the parameters are chosen to make the approximation as accurate as possible [4]. In particular, the so-called *variational* parameters are found by minimizing the Kullback-Leibler (KL) divergence between the tractable distribution and the true posterior in eq. (4).

Figure 3 shows the graphical model that represents the variational approximation for the posterior in eq. (4). In essence, for each document, the true posterior $p(\theta_d, z_d|\mathbf{w}_d, \alpha, \beta)$ is approximated by the factorized form:

$$q(\theta_d, \mathbf{z}_d|\gamma_d, \phi_d) = q(\theta_d|\gamma_d) \prod_{n=1}^{N} q(z_{d,n}|\phi_{d,n}), \tag{5}$$

where $q(\theta_d|\gamma_d)$ is a Dirichlet distribution with parameter vector γ_d and $q(z_{d,n}|\phi_{d,n})$ is a multinomial distribution with probabilities $\phi_{d,n}$. Up to an additive constant, it is possible to compute the KL divergence between $q(\theta_d, \mathbf{z}_d|\gamma_d, \phi_d)$ and $p(\theta_d, z_d|\mathbf{w}_d, \alpha, \beta)$ in terms of the variational parameters γ_d and $\phi_{d,n}$. The variational parameters are found by iteratively minimizing this expression by some form of gradient descent.

Variational methods not only play an important role in inference, but also in parameter estimation [4, 18]. In particular, they provide a lower bound on the log likelihood of a document, given by:

$$\log p(\mathbf{w}_d|\alpha,\beta) \geq \mathrm{E}_q \left[\log \frac{p(\theta_d, \mathbf{z}_d, \mathbf{w}_d|\alpha,\beta)}{q(\theta_d, \mathbf{z}_d|\gamma_d, \phi_d)} \right], \tag{6}$$

where E_q denotes an expectation with respect to the variational distribution. Let $L(\gamma_d, \phi_d; \alpha, \beta)$ denote the lower bound on the right hand side of eq. (6). In practice, the parameters α and β are estimated by maximizing this lower bound summed over all documents in the corpus:

$$(\hat{\alpha}, \hat{\beta}) = \arg\max_{\alpha,\beta} \sum_{d=1}^{D} L(\gamma_d, \phi_d; \alpha, \beta) \tag{7}$$

The learning procedure in this framework is known as a variational EM algorithm. The E-step computes the variational parameters for each document that minimize the KL divergence between eqs. (4) and (5). The M-step computes the parameters (α, β) that maximize the overall lower bound in eq. (7). The two steps are iterated until convergence. The overall procedure can also be viewed as a double (alternating) maximization of the lower bound on the log-likelihood in terms of the variational and model parameters for LDA. Finally, note that once a model is trained, we can approximately infer the topic proportions of each document from $\mathrm{E}_q[\theta_d|\gamma_d]$ and the topic assignments of words from $\mathrm{E}_q[\mathbf{z}_d|\phi_d]$.

4. EVALUATION

In this section, we apply LDA to the Freelancer data set. First, we describe how we preprocess the data set and select parameters for LDA. Then, we investigate the LDA results, starting with the clusters of abuse jobs LDA identifies and the keyword sets that characterize them. We then compare the results of using LDA with previous results that used a supervised approach.

4.1 Data Set Preprocessing

The data set includes information about both projects and users. Each project has a title, description, keyword phrases (manually selected by buyers from a predefined list), and the posting time (see Figure 5 for an example). The self-selected keyword phrases are not useful by themselves, as they are typically vague and do not separate the different job classes. For users, we have background information (country of origin, skill sets, etc.) on the buyers who commissioned projects and the workers who bid on projects.

We construct documents from the title, description and keywords of a project. LDA does not attempt to model word ordering; hence we represent the data by a term-document matrix whose elements simply count how many times each term occurs in each document (i.e., job posting). For the project title and description, we lowercase words, split at punctuations, remove stopwords, and apply stemming. We use the stopword list from [19], and append some corpus-specific stopwords to the list such as ‘wanted’, ‘bid’ and ‘freelancer’. We do not process the buyer-selected keywords: by leaving them capitalized, we can discriminate between keywords and preprocessed terms.

After processing the text we apply several filters to the data set. We only include the projects of “active” buyers in our analysis, where “active” is defined as buyers involved in twenty or more projects. This is done to guarantee that each buyer has a sufficient number of projects to estimate their topic proportions. We apply a similar filter to workers, removing those who made less than twenty bids. Finally, we remove terms that occur in less than six projects; this reduces the size of the dictionary and makes the learning more efficient. (These terms do not contain much information as topics are mainly identified by terms that appear in many documents.) After applying these filters, we obtain a term-document matrix with 27,600 terms and 355,386 documents.

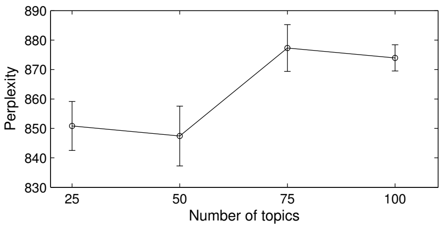


Figure 4: The perplexity over different number of topics. Each point is the average from the five samples, and the error bar denotes standard deviation.

4.2 Setting LDA Parameters

For the experiments, we use LDA with a symmetric Dirichlet distribution, initializing α to 1. Recall that each row of β corresponds to a word distribution for each topic. To seed the algorithm, we follow the initialization technique of [2]: for each topic, we choose five random documents and smooth their aggregated word counts by adding one to the count of each word in the dictionary.

One goal of our work on topic modeling is to discover the number of distinct project types, parameterized by K. We can estimate the number of project categories by examining how topic models of different sizes generalize to project descriptions that are not included as training data [2]. The generalization performance is measured by the so-called perplexity [4]:

$$\text{perplexity}(M_{\text{validation}}) = \exp\left[-\frac{\sum_{d=1}^{D} \log p(\mathbf{w}_d|\hat{\alpha}, \hat{\beta})}{\sum_{d=1}^{D} N_d} \right], \tag{8}$$

where the sum is over all project descriptions in a separate validation set—i.e., those documents that have been purposely withheld from the training set in order to validate our choice of the number of topics. Note that the perplexity is a decreasing function of each model’s likelihood on the validation set; hence smaller perplexity means better generalization. Admittedly, as pointed out by [7], the held-out perplexity does not always indicate a number of topics consistent with human evaluations. An alternative interpretation is that the held-out perplexity indicates the number of topics that can be reliably discovered based on the nature and amount of training data that is available.

To choose the number of topics for our experiments, we conducted five runs of LDA for each $K \in \{25, 50, 75, 100\}$ on the training set (90% of documents), and measured the perplexity on the validation set (remaining 10%). Of these, Figure 4 shows that K=50 yielded the best perplexity; thus, we used this parameter value when applying LDA on the entire data set.

4.3 Discovered Topics

We start exploring the results from topic modeling by examining the matrix elements of β. Recall that the largest elements in each row of β indicate the most probable words in each of the model’s discovered topics. The most probable words, however, do not always reveal the differences between several related but distinct topics. For example, the word “google” might occur with high probability in multiple “SEO” topics, but it does not differentiate various topics under “SEO”, such as “SEO Whitehat” and “SEO Greyhat”. Blei and Lafferty [2] suggested the following score to

No.	Title	Ratio	Top 20 Terms with Highest Scores
1	SEO Content Generation	4.66%	articl writer Articles write **copyscap** 'Article Rewriting' Copywriting word english grammar Ghostwriting nativ rewrit sampl plagiar grammat topic Blog spell re-writ
2	SEO Content Generation	3.16%	articl keyword rewrit **copyscap** word rewritten Copywriting paragraph sentenc topic content writer phrase densiti origin english grammat plagiar write research
3	SEO Whitehat	2.94%	link pr site page anchor websit nofollow farm **googl** robots.txt 'Link Building' text cloak backlink ip directori perman redirect web non-reciproc
8	CAPTCHA Solving	2.54%	'Data Entry' data 'Data Processing' entri team captcha Excel fast worker hr pm **indian** BPO 'Virtual Assistant' night id Research **india** 'Web Search' pandeypriya
13	Clicks/CPA/Leads/Signups	2.31%	market sale traffic promot affili lead 'Internet Marketing' Marketing commiss Sales Advertising Telemarketing campaign Leads sell telemarket 'Bulk Marketing' month earn busi
14	Ad Posts/Accounts	2.29%	ad account **craigslist** post pva poster **gmail** **cl** Freelance ip 'Data Entry' proxi ghost day citi flag classifi 'Classifieds Posting' 'Data Processing' daili
20	SEO Unknown	2.08%	seo keyword **googl** search rank engin SEO optim 'Link Building' adword result meta adsens 'Internet Marketing' traffic top tag analyt ppc **yahoo**
24	Bulk Emailing	1.88%	email list address excel mail newslett e-mail spreadsheet contact collect send sheet fax scrape inbox bulk outlook number mass smtp
30	OSN Linking	1.74%	fan **facebook** member profil friend **Facebook** **myspac** **twitter** account event bot membership page invit **Twitter** 'Social Networking' suspend group real **fb**
33	SEO Greyhat	1.71%	blog post forum comment 'Link Building' thread phpbb poster vbulletin 'Forum Posting' Blog dofollow **blogger** SEO topic signatur 'Internet Marketing' irfan board spam
40	SEO Greyhat	1.55%	submiss directori review social bookmark submit copi network **dmoz** 'Link Building' list SEO **digg** manual 'Internet Marketing' media press **squidoo** submitt past
42	Clicks/CPA/Leads/Signups	1.43%	sign signup citi countri **uk** up **usa** **canada** travel adult **australia** state unit restaur real station south region **europ** club

Table 2: The abuse-related topics discovered by LDA. The first column 'No.' is the index of clusters from LDA (sorted by ratio), and the third column 'Ratio' denotes the ratio of the number of projects in the cluster to the total projects in the filtered data set. The ratio is computed from the soft clustering where each project has positive proportions over multiple clusters. We title each cluster, and list the top twenty terms according to the score shown in eq. (9). Bold words denote references to companies or countries; underlined words underscore terms that shed light into worker methodologies; capitalized words are job keywords.

identify distinctive keywords in each topic:

$$\text{term-score}_{k,v} = \hat{\beta}_{k,v} \log \left[\frac{\hat{\beta}_{k,v}}{\left(\prod_{j=1}^{K} \hat{\beta}_{k,v} \right)^{\frac{1}{K}}} \right] \tag{9}$$

Intuitively, the score in eq. (9) highlights words that appear with high probabilities in one or a few topics and low probabilities in all the others.

We inspected the top scoring terms and manually examined random projects assigned to the fifty topics. Ultimately, we identified twelve abuse topics; the remaining topics were for "benign" jobs such as Web programming tasks. Table 2 shows the discovered abuse-related topics along with their top 20 scoring terms. We assigned a label to each cluster based on the categories described in [22]. The top scoring terms are interesting for a variety of reasons: not only can they be used to quickly identify possible abuse jobs via simple text searching, but they also provide insight into the targeted Web services, the methodologies used, and the related dependencies among abuse jobs.

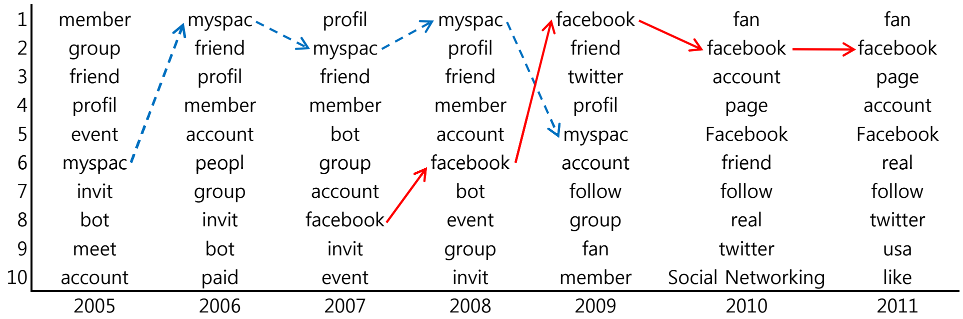
For instance, "SEO Content Generation" jobs seek workers to generate keyword-rich text that attracts page views but does not necessarily provide high-quality content; the goal of these jobs is to manipulate page rankings in search engine results. Freelancer buyers often specify a certain density of keywords (densiti), and they check the originality of the work using CopyScape (**copyscap**). The other two specific SEO jobs,"SEO Greyhat" and "SEO Whitehat", are backlink-based, meaning that the buyers want to increase the number of Web sites linking to their own. Greyhat jobs allow for such abusive tactics as spamming blogs (**blogger**), social bookmarking sites (bookmark, **digg**), and forums with backlinks. Whitehat jobs explicitly forbid these tactics, instead requesting backlinks from sites with particular page ranks (pr) but do not specify how to obtain them.

The practice of hiring human labor to circumvent the CAPTCHA defense mechanism is now commonplace [21]. For "CAPTCHA Solving" jobs, the buyers typically want Indian workers (**india**) who can work nighttime shifts (night, pm) solving CAPTCHAs. The "Ad Posts/Accounts" topic includes two closely related job classes. In the first, "Ad Posting", buyers pay workers to spam Craigslist (**craigslist**, **cl**) and other classified sites with daily advertisements, in an effort to achieve better placement in search results. However, to post on many sections in Craigslist, one needs a phone-verified account (pva), and these accounts are often created in conjunction with Web-based email accounts (**gmail**). Thus, abuse jobs for ad posting often build on top of those for account registration, which likely explains why the topics are merged.

"Online Social Network (OSN) Linking" is a relatively new abuse job focused on spamming users with advertisements by "friending" them or having them "like" pages. Table 2 shows that these jobs target specific social networks (**Facebook**, **twitter**), and buyers want page likes (page) and friends (friend).

4.3.1 Word Topic Assignments

Now that we have analyzed the discovered topics, we discuss the example project shown in Figure 5. The example illustrates the use of LDA on a concrete job posting; it is particularly interesting because this job covers multiple abuse tasks, as evident in its top three topic proportions: "SEO Greyhat", "Ad Posts/Accounts", and "SEO Content Generation". The buyer for this job requires accounts on various blog sites and wants the accounts populated with low quality content about car finance. The buyer might be attempting to increase the page rank for a specific car financing Web site or to monetize the blog accounts through online advertisements.



	2005	2006	2007	2008	2009	2010	2011
1	member	myspac	profil	myspac	facebook	fan	fan
2	group	friend	myspac	profil	friend	facebook	facebook
3	friend	profil	friend	friend	twitter	account	page
4	profil	member	member	member	profil	page	account
5	event	account	bot	account	myspac	Facebook	Facebook
6	myspac	peopl	group	facebook	account	friend	real
7	invit	group	account	bot	follow	follow	follow
8	bot	invit	facebook	event	group	real	twitter
9	meet	bot	invit	group	fan	twitter	usa
10	account	paid	event	invit	member	Social Networking	like

Figure 6: Top 10 keywords in projects assigned to the "OSN Linking" topic from 2005–2011. Note how Myspace was originally the most highly targeted OSN for abuse; since 2010, however, Facebook and Twitter have become the dominant targets.

Title: Open 10 blog *accounts* and write/publish 10 posts
Description: I need someone to open a free email *account* (i.e *yahoo*, *hotmail*, *gmail*)
Then use that email to open 20 free blog *accounts* (excluding blogger, word press, blog) This project is to open 20 free blog *accounts* (all different sites) than post a *single* blog post (20 in total) one each blog *account*. Detailed nformation below.
Each free blog *account* to be on a separate free blog service.
If you do not have good knowledge of where these free blogs are, do not bid.
These free blog *accounts* can be anywhere in the world as long as English speaking. Each blog must not be against any rules of these blogs.
Each blog *title* will be related to car finance
I then need you to write one article on each blog (20 in total) of at least 250 words regarding car finance.
I then will need a spreadsheet with user names and *passwords* of the free email *account* and all 20 blogs, so I can continue with the development.
Keywords: Blog

Figure 5: An example of assigning topics to each word in a specific project. We differentiate the topics in the text as follow: topic number 33 ("SEO Greyhat"), 14 ("*Ad Posts/Accounts*"), 1 ("SEO Content Generation") and 24 ("Bulk Emailing"). The proportions of these topics are 0.276, 0.211, 0.169 and 0.082, respectively.

For each project, LDA assigns words to specific topics; here, we see the terms *account*, *password*, and *gmail* linked with the "Ad Posts/Accounts" topic. The first step in spamming blogs is to obtain accounts on blogging services; these services typically require valid email accounts. Next, we see that the keywords blog and post are assigned to "SEO Greyhat", words that are strongly associated with abusive techniques for achieving backlinks. Lastly, words like article and write are associated with "SEO Content Generation".

LDA is not perfect, however, and this example also illustrates some of LDA's limitations. LDA does not attempt to model word ordering; it only models overall counts of word occurrences in the document. Presumably this is why words like email (more properly assigned to Account Creation in this context) and spreadsheet cause the job to be associated with "Bulk Emailing", even though this posting has nothing to do with that task.

4.3.2 Keyword Trends

The keywords LDA identifies for each topic also reveal interesting trends over time. Figure 6 shows the top 10 keywords that appear in projects assigned to the "OSN Linking" topic from 2005 to 2011. We focus on this topic as the OSN landscape has undergone several significant changes over time. We see that Myspace was originally the most highly targeted OSN from 2005–2008, which corresponds to the era when Myspace was the dominant social networking platform [9]. At that time, buyers wanted workers to obtain friends and group members (friend, group, member, invit) for their Myspace accounts (myspac, account). Also, buyers would post job descriptions to create automated friending software (bot); before 2009, buyers were willing to assemble the necessary software tools to circumvent defense mechanisms targeting automated programs. During the rise of Facebook and Twitter in 2009, buyers began focusing their efforts on them (facebook, twitter), requesting friends, fans and followers (fan, follow). By 2010, Myspace disappeared from the top 10 list. Facebook and Twitter remain the most highly targeted OSNs for abuse, as buyers continue to commission jobs to gather "likes" for pages (like, page) and "followers" from real people based in the U.S. (real, usa).

4.4 Comparison to Supervised Learning

Next we consider how to validate our interpretation of the topics discovered by LDA. To do so, we compare our results from LDA to those obtained by supervised learning in previous work.

4.4.1 Topics

In [22], 146,657 job postings were classified into 10 different abuse categories by support vector machines (SVMs) trained on 10,978 manually labeled examples. The intersection of the classified postings with our filtered data set yields 67,963 postings. Each of these postings can therefore be viewed as having a class label c (obtained from the SVMs) and a dominant topic t (as revealed by the topic in LDA with the highest inferred proportion). We can measure the correlation of the results from supervised and unsupervised learning by asking how well, for each of these postings, the dominant topic t predicts the class label c.

More formally, we measure the correlation between these results by computing the reduction of uncertainty in the class label c given the dominant topic t. The uncertainty coefficient [24] measures this reduction by a ratio of conditional and unconditional entropies. Specifically, we regard C and T as random variables denoting the class labels (from supervised learning) and dominant topics (from unsupervised learning). We estimate the joint probability $p(c, t)$ by counting the number of abuse job postings with class label c and dominant topic t, then dividing by the total number of abuse job postings. From the joint probability $p(c, t)$ estimated in this way, it is straightforward to obtain the marginal probability $p(c)$ and the conditional probability $p(c|t)$. In terms of these probabilities, the

	Top 3 Topics					
Class	No.1	Ratio	No.2	Ratio	No.3	Ratio
Verified Accounts	14	89.5%	30	2.3%	44	2.2%
SEO Content Gen.	1	52.0%	2	34.6%	10	2.4%
SEO Whitehat	3	97.0%	20	0.8%	37	0.6%
SEO Greyhat	33	33.3%	40	25.1%	3	20.9%
Account Regi.	14	65.3%	30	8.7%	8	5.6%
CAPTCHA	8	94.0%	14	0.8%	5	0.8%
Ad Posting	14	87.4%	33	6.3%	13	1.1%
OSN Linking	30	83.4%	34	5.0%	13	3.0%
Bulk Emailing	24	74.2%	13	13.4%	18	2.4%
Signups	13	52.7%	42	25.3%	9	4.2%

Table 3: The assignment of projects in each class of supervised learning to discovered topics.

uncertainty coefficient is given by:

$$U(C|T) = 1 - \frac{H(C|T)}{H(C)} = 1 - \frac{\sum_{c,t} p(c,t) \log p(c|t)}{\sum_{c} p(c) \log p(c)}, \quad (10)$$

where $H(C)$ is the entropy of C and $H(C|T)$ is the conditional entropy of C given T. The uncertainty coefficient in eq. (10) is bounded between 0 and 1: a value of 0 indicates that the two variables are entirely uncorrelated (as if the value of c, given t, was drawn completely at random), while a value of 1 indicates that the first variable is completely determined by the second one. For the (supervised) class labels c and (unsupervised) topics t, we obtain an uncertainty coefficient of 0.719. This value indicates a significant correlation between the results of classification using 10,978 manually (and meticulously) labeled job postings and topic modeling from 355,386 unlabeled job postings (of which many more are easily acquired)

For more detail, Table 3 shows the assignment of projects in each class — from supervised learning in [22] — to our discovered topics. The assignment is consistent with our analysis of the discovered topics, although some of the manually labeled classes were split in two (class "SEO Content Generation"), while others were combined to form one topic (class "Verified Accounts", "Account Registration" and "Ad Posting"). Each row shows the percentage of jobs from a supervised class that appeared in a corresponding unsupervised cluster. Topics with very distinctive words like "CAPTCHA Solving" produced results on par with the supervised approach: 94% of the projects classified as "CAPTCHA solving" class using SVMs were placed in the "CAPTCHA solving" cluster.

4.4.2 User Profiles and Topic Correlations

Next we use the clusters from LDA to profile the population of workers who bid on abuse jobs and the population of buyers who post their project descriptions. We also examine the correlations among these user profiles, which serves as one method for discovering mergeable topics.

First, we calculate user job profiles based on topic proportions. In the Freelancer data set, each job posting (i.e., project title, description, and keywords) is written by one buyer, and subsequently bid on by one or more workers. In addition to estimating topic proportions for each posting (as revealed by the latent variable θ_d), we can also estimate topic proportions for individual buyers and workers. To profile buyers and workers in this way—and to explore the correlations between different types of buyers and workers—we adopt the following simple heuristic. Recall that $\mathrm{E}_q[\theta_d|\gamma_d]$ indicates the (approximately) inferred topic proportions for the dth posting in the corpus. Let $\mathcal{M}$ indicate the set of job postings associated with one user's history on Freelancer (i.e., either the set of all jobs written by a buyer, or the set of all jobs bid on by a worker). We estimate the topic proportion ω of that user as:

$$\omega = \frac{1}{|\mathcal{M}|} \sum_{d \in \mathcal{M}} \mathrm{E}_q[\theta_d|\gamma_d]. \quad (11)$$

Intuitively, the right hand of eq. (11) estimates the user topic proportions by simply averaging over the inferred topic proportions of all the user's job postings.

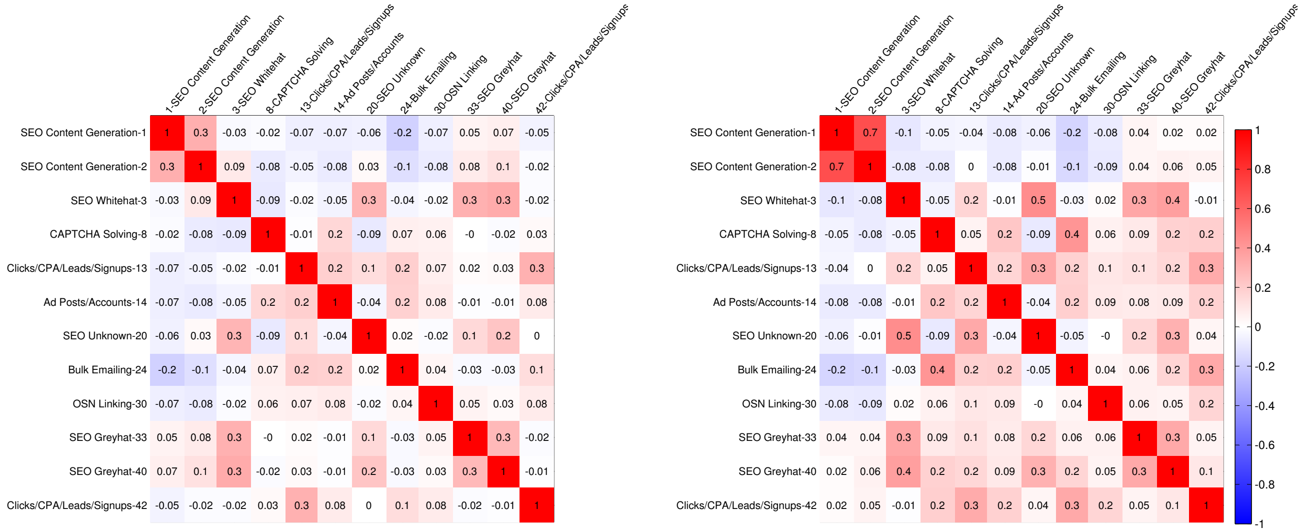
Having profiled buyer and workers based upon their topic proportions, we can then explore inter-topic relationships by calculating the Pearson correlation matrices for the buyer and worker topic proportions. The element in row i and column j of these matrices is a coefficient, bounded between -1 and 1, that measures the linear dependence of the ith and jth topic proportions. Coefficients with large magnitudes indicate a strong linear relationship between different topic proportions, which can be either positively or negatively correlated. Conversely, coefficients with small magnitudes indicate a weak linear relationship.

Figure 7 shows the correlation matrices for the buyers and workers. These matrices allow us to reason about related topics from both the buyer and worker perspectives. Note that these matrices are markedly different from the overall document correlation matrix (which computes correlations across all job postings, as opposed to those linked to a particular user); we do not show the overall document correlation matrix because the off-diagonal entries are largely close to 0. For example, topics 1 and 2 in the document matrix have a correlation coefficient of 0.1, but have values of 0.3 and 0.7 in the buyer and workers matrices.

The correlation matrices provide both insight into the data (revealing typical buyer and worker job profiles) as well as our methodology (revealing LDA limitations). In terms of job profiles, Figure 7 shows weak correlations among both the buyers and workers of the related link-based SEO jobs ("SEO Whitehat", "SEO Greyhat", and "SEO Unknown"), as well as account creation ("Ad Posts/Accounts") and subverting account defenses ("CAPTCHA Solving"). Other jobs, in particular buyers for "OSN Linking" jobs, are in their own niche completely separate from the other topics.

The correlation matrices also reveal two limitations of LDA. First, some of the correlated topics in practice can be merged into a single topic. Figure 7 shows strong correlations between topics 1 and 2 (from Table 2), 13 and 42, and 33 and 40. Further manual inspection shows that these topics are in fact quite similar (which we reflect in their topic titles), differing only slightly in their keyword sets. For example, even though LDA separated topics 1 and 2 from each other, they in fact are related. The job postings in these topics mainly differ in the use of specific terms: for instance, job postings in the "SEO Content Generation" class that include 'Blog' as a keyword fall into topic 1, while those that have more detailed requirements (using the words 'paragraph', 'sentence' and 'phrase') fall into topic 2. The buyer correlation value provides us with a weak indication that topics 1 and 2 are related. The signal is weak because buyers tend to use the same terms to describe their projects; if a buyer posts a project heavily proportioned to topic 1, then any future postings relating to "SEO Content Generation" are likely to repeat the same terms when the buyer recycles old project descriptions. Workers, however, recognize that jobs from both topics are the same (as reflected in the much stronger correlation of 0.7 of the two topics in Figure 7(b)).

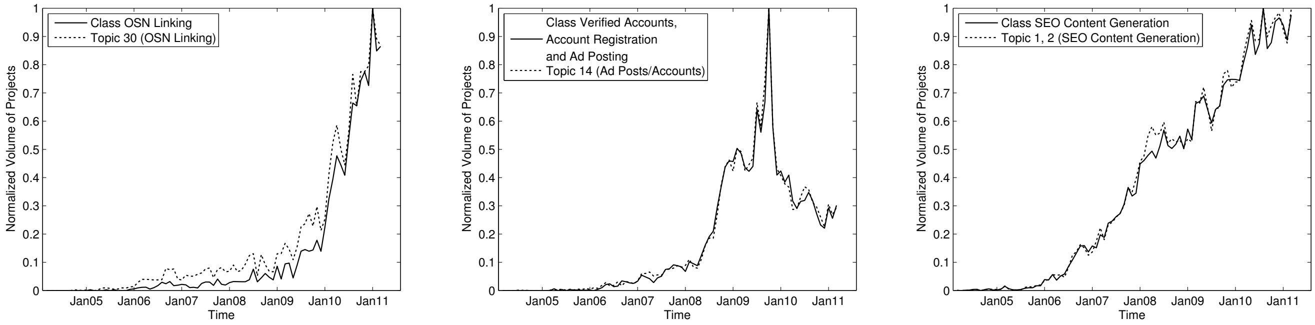
Second, the correlation matrices also show false relationships between topics. For example, the results show a strong correlation between "Bulk Emailing" and "CAPTCHA Solving". Upon inspecting projects with topic proportions greater than 0.3 for both top-



(a) Buyers

(b) Workers

Figure 7: The correlation matrix for buyer and worker topic proportions.



(a) OSN Linking

(b) Ad Posting/Account Creation

(c) SEO Content Generation

Figure 8: Demand for several representative topic trends over time. We assign jobs to the topic with the highest proportion, then bin according to month. The bins are normalized using the maximum monthly value across all time. The dotted lines correspond to the LDA clustered jobs, while the solid lines are taken from [22].

ics, we determined that the correlation exists due to false positives. Both "Bulk Emailing" and "CAPTCHA Solving" were assigned projects consisting predominantly of private jobs. A private job is one where the buyer directly addresses a project to a certain worker, typically done by writing "Private job for <username>" without a detailed description. The most informative terms in the projects shared across both topics consisted of such user-selected keywords as 'Data Entry' and 'Data Processing'.

4.4.3 Topic Trends

Figure 8 shows the evolution of demand for several representative topics over time, together with the same trends observed using the supervised methods from previous work [22]. Both techniques offer similar views on how demand fluctuates for each job class. These figures provide further evidence that LDA properly captures the abuse-related job classes, as the trends in demand track each other closely. LDA places a greater number of jobs into "OSN Linking" topic than the classifier of previous work [22] did. Previous work found that 1.3% of jobs fell into this category, while LDA produced a value of 1.7%. Randomly sampling among the postings in that cluster shows that jobs involving the creation of Facebook apps and OSN Web site clones were incorrectly included in this topic. We also see that LDA does not always offer as fine-grained a separation of job classes, as illustrated by Figure 8(b). Only after combining three different job classes from the supervised method do we observe a similar demand trend. Lastly, the "SEO Content Generation" cluster exhibits very little difference with the supervised approach, as shown in Figure 8(c). Though we merged two clusters together to generate the demand curve, LDA produced results on par with supervised methods for classification.

5. DISCUSSION

Attackers are increasingly turning to cheap, human labor to abuse popular Web services. The manpower for these attacks depends on Web sites, such as Mechanical Turk and Freelancer, that connect attackers with potential workers. In this paper, we have explored a new approach that freelancing sites might adopt to identify and monitor job postings related to abuse. Our approach is an alternative to previous work in supervised learning. It is based on LDA, an unsupervised method for topic modeling, that can effectively cluster free-form job postings. LDA provides a largely automated method for recognizing postings related to abuse. Below we discuss its main strengths and weaknesses, as well as opportunities for future work.

One strength of LDA is that it yields fairly interpretable clusters of job postings. The top-weighted words in each topic not only identify the clusters of jobs related to abuse. They also provide valuable insight into the major Web service targets (e.g., Gmail, Facebook) and the methodologies for executing tasks (e.g., proxies, keyword densities).

Another strength of LDA is that it eliminates the need for an initial manual labeling of job postings. Previous work in classification of abuse jobs required a time-consuming, manual labeling of thousands of job postings. By contrast, without any manual labeling, we used LDA to perform a clustering of over ten times as many job postings. Our results suggest that LDA can be deployed at scale and used by freelancing sites without a large investment of human resources.

LDA does, however, have some limitations compared to supervised approaches. For the Freelancer data set, some LDA clusters lacked the same granularity as those obtained from manual labeling; this occurred when 2–3 related but distinct job categories were merged into a single topic. Also, sometimes LDA split single job categories into two topics—not because the underlying jobs were different, but because they were described in different ways. To address these issues, we explored ways for identifying mergeable topics, most notably by examining the correlation matrices of topic profiles for buyers and workers. Anecdotally, LDA also tends to have more false positives, an issue for further work.

Going forward, we see many potential applications of LDA in related areas, especially when analyzing the unstructured text that commonly appears in underground market interactions. Franklin et al. [11] analyze the content of an IRC channel involved in the exchange of illicit goods. They manually label thousands of advertisements using eight pre-chosen labels; LDA can be used to improve both aspects of the method. Stone-Gross et al. [25] describe a brief analysis of the underground forum `spamdot.biz`. They do not perform an in-depth investigation of the forum, perhaps due to the difficulty handling free-form text. Again, we see an application for LDA to extract meaningful topics from the forum posts.

Finally, the success of LDA on our data set suggests the application of more sophisticated topic models to freelance job postings. For example, in this paper we did not exploit the manually labeled job postings available from previous work. However, class labels can be incorporated into a supervised variant of LDA [1], biasing the model to discover latent topics that are predictive of the known labels. Further, dynamic topic models [3] could incorporate the date that jobs were commissioned and trace the evolution of different categories of abuse jobs over time. We hope to probe the social network of buyers and bidders by developing an extension of LDA such as [6]. This extension would not only model the identities of those who post and respond to abuse job solicitations, but also the connections between them. Even more recently, the demand for large-scale applications has led to work on an online version of LDA [14]; such an approach could be adapted for the continuous modeling of streaming projects on freelancing Web sites.

6. ACKNOWLEDGMENTS

We thank Stefan Savage and Kirill Levchenko for insightful comments and discussion of this work, and we also thank the anonymous reviewers for their valuable feedback. This work was supported in part by National Science Foundation grants NSF-0433668 and NSF-0831138, by the Office of Naval Research MURI grant N000140911081, and by generous research, operational and/or in-kind support from Google, Microsoft, Yahoo, Cisco, HP and the UCSD Center for Networked Systems (CNS). Kim was supported by a Samsung Scholarship.

7. REFERENCES

[1] D. Blei and J. McAuliffe. Supervised topic models. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 121–128. MIT Press, Cambridge, MA, 2008.

[2] D. M. Blei and J. Lafferty. Topic Models. In *Text Mining: Theory and Applications*. Taylor and Francis, London, UK, 2009.

[3] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, page 113–120, Pittsburgh, Pennsylvania, 2006.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3:993–1022, Mar. 2003.

[5] J. Chang and D. M. Blei. Hierarchical relational models for document networks. *The Annals of Applied Statistics*, 4(1):124–150, Mar. 2010.

[6] J. Chang, J. Boyd-Graber, and D. M. Blei. Connections between the lines: augmenting social networks with text. In *Proceedings of the Fifteenth ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 169–178, Paris, France, 2009.

[7] J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, and D. Blei. Reading Tea Leaves: How Humans Interpret Topic Models. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, page 288–296. 2009.

[8] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391—407, 1990.

[9] Facebook Overtakes Myspace. `http://blog.alexa.com/2008/05/facebook-overtakes-myspace_07.html`.

[10] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, volume 2, pages 524– 531 vol. 2. IEEE, June 2005.

[11] J. Franklin, V. Paxson, A. Perrig, and S. Savage. An Inquiry into the Nature and Causes of the Wealth of Internet Miscreants. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Alexandria, VA, Oct. 2007.

[12] Freelancer.com. `http://www.freelancer.com/info/about.php`.

[13] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, Apr. 2004.

[14] M. Hoffman, D. Blei, and F. Bach. Online learning for latent dirichlet allocation. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 856–864. 2010.

[15] T. Hofmann. Probabilistic Latent Semantic Indexing.

Research and Development in Information Retrieval, pages 50–57, 1999.

[16] D. J. Hu and L. K. Saul. A probabilistic topic model of unsupervised learning for musical-key profiles. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, 2009.

[17] P. G. Ipeirotis. Analyzing the Amazon Mechanical Turk Marketplace. *XRDS: Crossroads*, 17:16–21, Dec. 2010.

[18] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An Introduction to Variational Methods for Graphical Models. *Mach. Learn.*, 37(2):183–233, Nov. 1999.

[19] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. SMART stopword list. `http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop`, April 2004.

[20] A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research*, 30:249–272, Oct. 2007.

[21] M. Motoyama, K. Levchenko, C. Kanich, D. McCoy, G. M. Voelker, and S. Savage. Re: CAPTCHAs — Understanding CAPTCHA-Solving from an Economic Context. In *Proceedings of the USENIX Security Symposium*, Washington, D.C., Aug. 2010.

[22] M. Motoyama, D. McCoy, K. Levchenko, S. Savage, and G. M. Voelker. Dirty Jobs: The Role of Freelance Labor in Web Service Abuse. In *Proceedings of the USENIX Security Symposium*, San Francisco, CA, Aug. 2011.

[23] H. Ning, Y. Hu, and T. S. Huang. Searching Human Behaviors using Spatial-Temporal words. In *IEEE International Conference on Image Processing (ICIP)*, volume 6, Oct. 2007.

[24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 2007.

[25] B. Stone-Gross, T. Holz, G. Stringhini, and G. Vigna. The Underground Economy of Spam: a Botmaster's Perspective of Coordinating Large-Scale Spam Campaigns. In *Proceedings of the 4th USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET)*, Apr. 2011.