As Webpages Get Narrower, Do Ads Get Nichier?
An Online Field Experiment in Google Contextual Ads

Note: 1st Paper

Stephen Bruestle*

April 2014

Abstract

Firms target their advertisements to the consumer segments delivered by webpages. I develop an auction model where firms target segments of heterogeneous consumers. From this, I derive an empirical framework, which I use to test whether more niche or more general ads win the auction for more narrowly-focused webpages. To do this, I create many differentiated webpages in an experimental fashion and observe the Google text ads that are placed on them. Then, I compare general webpages, such as a 'Ford' webpage, with more narrowly-focused webpages, such as a 'Ford Truck' webpage, by using a measures of ad niche-ness. I use a Hierarchical Latent Dirichlet Allocation algorithm from the machine-learning literature to create a robust measure of ad niche-ness. My results show a U-shaped relationship between webpage narrowness and ad niche-ness: Ads for less niche products tend to appear on moderately narrow-focused webpages.

JEL Codes: M37, C93, C81, D22, L25

KEYWORDS: online field experiment, Google textual targeted advertising, topic modeling, hierarchical latent dirichlet allocation, niche ads, narrow-focused webpages

*sdb8g@virginia.edu (email), (609) 540 - 1861 (phone), http://stephen.bruestle.net/ (website). I thank Federico Ciliberto, Simon Anderson, Nathan Larson, Steve Stern, Dina Guo, Allan Collard-Wexler, Marius Schwartz, Olivier Armantier, and Paris Cleanthous for their helpful comments. I also thank the participants of several workshops at the University of Virginia (November 2012; September 2012; December 2011) and Lafayette College (November 2011) for their invaluable comments. I also thank the UVACSE tiger team program and Katherine Holcomb for invaluable programming and harware services. I also thank John, Cary, Thomas, and Anna Bruestle (my family) for their invaluable programming assistance and loving support. In addition, I thank the Bankard Fund for Political Economy for its generous financial support.

1 Introduction

Whenever we update our statuses on Facebook, search for videos of cats on YouTube, or email our loved ones on GMail, we see advertisements targeted toward our habits. Our cyber-world is teeming with *targeted advertisements*where different ads are shown to different consumers based on their tastes, locations, or demographics. While much technology and innovation has improved advertisers' abilities to select to whom they send their ads, some of the webpages on which they advertise continue to draw in large, heterogeneous segments of consumers. While there are webpages on narrow topics, such as a webpage on Texas Hold'em Poker strategies, there are many webpages on broad topics (such as webpages dedicated toward card games in general or main index pages linking to more narrowly-focused webpages). Advertisers bid for ad space on both these more generally-focused and more narrowly-focused webpages.

The ads that win the online auction for a more narrowly-focused webpage may not be for niche products. To keep my language simple, I will refer to ads for more niche products as *more niche ads* . For example, what if no niche products exist that could cater to those who would visit a more narrowly-focused webpage? Would a close product win the auction, or would a more general one succeed? For example, while there are many products targeted toward Nascar drivers, like seatbelt harnesses and racing helmets, and there are many products targeted toward opera singers, like libretos and voice lessons, there may be few, if any, products targeted toward opera singing Nascar drivers. A more narrowly-focused webpage dedicated to Nascar drivers singing the most famous arias therefore may not get more niche ads than a webpage dedicated to Nascar drivers or opera singers. Instead, we could see on the webpage ads targeted toward opera singers, ads targeted toward Nascar drivers, or ads targeted toward a broader audience, such as ads for credit cards or insurance.

In this paper, I test whether ads for products targeted toward smaller market segments focused on marketing niche products or ads for products targeted toward larger

market segments focused on marketing general products would win the auction for more narrowly-focused webpages. For example, would a webpage featuring 'Ford Truck Tires' show more ads for Ford Truck snow tires (a more niche product than Ford Trucks) or more ads for Ford cars (a less niche product than Ford Trucks) than the parent 'Ford' webpage. The purpose of this paper is not examine Google's strategy, because I assume they are selling ad space in a simple auction, but to explore the targeting behavior of firms as a function of the preference of each consumer, which is a preference revealed by his visit to a particular webpage.

Despite the recent theoretical economic literature on targeted advertising (see for example: Bergemann and Bonatti, 2011; Iyer et al., 2005; Johnson, 2013), there has not been empirical analyse of targeted advertising. The one expectation is Chandra (2009). He uses the number of competing newspapers as a proxy for how targeted or niche advertising is in a city, which shows that more competing newspapers lowers circulation prices and raises advertising prices. This paper extends the discussion by examining the kinds of ads delivered to a more narrowly-focused audience.

I start this paper by developing a Hotelling model where firms bid on advertising toward segments (webpages) of heterogeneous consumers in a second-price auction. I use this model to investigate whether an ad for a product targeted toward a smaller market segment selling a niche product or an ad for a product targeted toward a larger market segment selling a general product would win the auction for more narrowly-focused webpages. Using an example, I demonstrate the possibility that the niche-ness of ads served on a webpage can vary non-monotonically with the narrowness of the webpage content. Because of the ambiguity of the theory, I test this empirically.

To test this empirical question, I run a novel field experiment on Google text-based, webpage advertisements. I create webpages that each contain: (1) unique webpage content and (2) a space for one Google text advertisement. For each webpage, Google observes the content and auctions advertising on that webpage based the perceived topic of the webpage.

My experiment is somewhat similar to two online field experiments run by Randall A. Lewis and David H. Reiley at Yahoo! Research (Lewis and Reiley, 2011, 2012). In these experiments, they identify a set of consumers in both Yahoo! and an online retailer's database. Then, they randomly assign these consumers different amounts of advertising. They test how changing online advertising behavior (how many ads a consumer sees) affects consumer behavior (purchasing decisions). My research likewise focuses on the complex subject of online advertising. In this experiment, I test how changing online consumer behavior (revealed to firms by the choice of a webpage) affects advertiser behavior (which firm wins the bid for advertising).

One advantage of creating my own webpages instead of using previously existing webpages or other advertising media, like real magazines or TV stations, is I can control their content. The ad slot value of a preexisting webpage presumably depends on a slew of variables, such as number of ads on a webpage or traffic, many of which are either unobservable to me or hard to measure and compare. By creating my own webpages, I can be sure that ad slot values depend on my content keywords and nothing else.

I chose to use content from websites that were currently being auctioned on Flippa. Flippa (flippa.com) is the largest online marketplace dedicated to buying and selling websites. For instance, nearly 26,000 websites were sold on Flippa in 2011, for a value of almost $31 million. The advantage of using websites currently auctioned on Flippa is that the seller provides data on the highest bid (i.e. the price for the website), age of the website, ad revenue, website traffic, website ranking, and many more variables. I did not use previously-auctioned websites despite that data being available, because the buyer could have changed the website's characteristics after the auction.

I then ran an automated program to observe advertisements on my webpages. My data collection program:[1] (1) randomly brings up one of my webpages in the firefox browser;

[1] I thank John, Thomas, and Anna Bruestle for their programming help on writing my data collection program. The program collects data from the firefox browser, not the source code, because Google programmers made it difficult, if not impossible, to get the ad directly from the source code. It was written in Javascript embedded into the webpages in a way such that Google would not be able to see the program.

(2) it grabs the webpage content, the text ad, the date, and the time off the webpage; (3) it randomly jumps to another of my webpages in Firefox. Then it iterating back on (2) to grab the information off the new webpage.

Intuitively, a Lincoln Blackwood, which is Ford's 2002-03 all-black luxury pick-up truck which sold only 3,356 units, is a niche product. In contrast, a Toyota Corolla, which is the perennial best selling car in the world, is not. Therefore, an ad for Lincoln Blackwoods would be more niche than an ad for Toyota Corollas. And a webpage on Lincoln Blackwoods would be more narrowly-focused than a webpage on Toyota Corollas. The methodological problem is finding some scientific and systematic way of measuring how niche an ad is and how narrow a webpage is. This is not trivial, because the measure needs to be exogenous. Therefore, I cannot use number of units sold, price of the product, or price of the ad as proxy variables for niche-ness and narrowness, because these are all endogenous measures.

While an ideal experiment might allow me to pick both the possible ads and the possible webpages, in this experiment, I cannot choose the possible ads. I instead observe real text ads created and bid on by real advertisers. Therefore, I cannot slightly vary the text in ads, to observe different levels of targeting. I cannot add the word 'red' or the word 'truck' to the text of an ad and see how it changes which webpages it gets posted on. Instead I observe the occurrences of different ads, some which contain the word 'red', some which contain the word 'truck', and some which do not contain either word. Therefore I create measures of advertisements to be able to identify which ads are similar and which ads are different.

I use a statistical method from the machine-learning literature to uncover the latent relationship in clusters of words to derive a measure for the niche-ness of an ad. This statistical method comes from a stream of machine-learning research known as *topic modeling* (see for example: Blei et al., 2003a; Griffiths and Steyvers, 2004; Minka and Lafferty, 2002; Teh et al., 2006b). Topic modeling algorithms are probabilistic algorithms for uncovering the underling structure of a set of documents using hierarchical Bayesian

analysis of the original texts. They are most often used to categorize documents based on observed patterns of words in the texts. This paper is the first economics application of a topic modeling algorithm.

In the past decade, most of the development of topic modeling has been from adaptations and applications of Blei et al.'s (2003a) Latent Dirichlet Allocation (LDA). This includes analyses of scientific abstracts (Griffiths and Steyvers, 2004) and newspaper archives (Wei and Croft, 2006). LDA is not only the most widely accepted topic model, but it is also the most powerful. A large part of the machine-learning literature has focused on creating faster and more efficient algorithms for estimating the latent relationship between words and documents documents using the LDA model, including mean field variational inference (Blei et al., 2003a), collapsed variational inference (Teh et al., 2006a), expectation propagation (Minka and Lafferty, 2002)), and Gibbs sampling (Steyvers and Griffiths, 2006). In this paper, I focus on applying a topic model algorithm to answer an economics question, not developing a new estimation technique nor develop a new topic model.

The big advantages of LDA are: It allows for documents to be generated from multiple topics, it allows the topics of documents to be identified without having to create a new exogenous variable for each document, and it can be identified quickly. Many of the algorithms based on LDA only take a few hours to run on a set of tens of thousands of documents. For a good overview of the important topic models and the importance of LDA, see Blei and Lafferty (2009).

In this paper, I focus on a particular adaptation of LDA from Blei et al. (2003b) called Hierarchical Latent Dirichlet Allocation (HLDA) because it allows me to test the nicheness and narrowness of documents (my ads and my webpages). HLDA imposes the additional assumption on the basic LDA model that the categories are hierarchical in nature. There is additionally one parent category, with a set of the most general words. Further, there is a set of children categories, with a set of more niche words. And there is a set of grandchildren categories, which are subcategories of the children categories and

are composed of the most niche words.[2] I use HLDA by using it to estimate the level of the category of each word in each ad; the lower the level, the more niche the word in the ad. Using this measure, I find strong evidence for a non-monotonic relationship: The less niche ads tend to appear more on moderately narrow webpages.

2 Hotelling Auction for Targeted Advertising

In this section, I develop a Hotelling model where firms bid on advertising toward segments (webpages) of heterogeneous consumers in a second-price auction. I use this model to investigate whether an ad for a product targeted toward a smaller market segment (niche product) or an ad for a product targeted toward a larger market segment (general products) would win the auction for a more narrowly-focused webpage. An advertiser for a niche product might bid more per customer for advertising on a narrow webpage that is targeted toward its small segment of potential buyers; or the advertiser might bid less for advertising on a narrow webpage that is targeted toward consumers that would not buy its product. Therefore, it is not obvious what kind of firm will tend to win the right to advertise on a narrower webpage.

I show an example where a general product wins the auction for advertising on a generally-focused webpage, a more niche product wins the auction for advertising on a more narrowly-focused webpage, and a general product wins the auction for advertising on the most narrowly-focused webpage.[3] I could have, however, just as easily have created a counter example where a niche product wins the auction for advertising on a generally-focused webpage, a general product wins the auction for advertising on a more narrowly-focused webpage, and a niche product wins the auction for advertising on the most narrowly-focused webpage. This shows that the relationship between the

[2] There can be as many levels to the tree as you choose, but I found that only three levels were useful for my dataset.

[3] In Appendix A, I extend this to a market equilibrium by introducing endogenous product prices and many firms.

niche-ness of the advertised product and the narrowness of the webpage is not necessarily monotonic. Therefore, I test it as an empirical question.

In 2.1, I first develop a base Hotelling model where firms bid on advertising seeking to attract individual heterogeneous consumers in a second-price auction. Then, in 2.2, I solve for the firm's bidding strategy where each firm bids its value for advertising.[4] Next, in 2.3, I extend my base Hotelling model so that firms bid on segments (webpages) of consumers, instead of on individual consumers. I use this analysis to lay the theoretical foundation for my empirical analysis.

2.1 Basic Model

There are two firms $j = 0, 1$ that each produce a horizontally differentiated good at a constant marginal cost, which is normalized to zero. Each firm j receives an i.i.d. random, horizontal product characteristic or location of $x_j \sim U[-1, 1]$, which is only known by firm j. For now, each firm j has an exogenous product price of p_j.[5] Each firm j simultaneously chooses its advertisement bidding function $b_j : \mathbb{R} \to \mathbb{R}_{\geq 0}$ to maximize its expectation of its profit $\Pi_j \equiv p_j \int_{\mathbb{R}} \omega_j(x)dx - \int_{\mathbb{R}} b_{1-j}(x)1\{b_j(x) > b_{1-j}(x)\}dx$, where $\omega_j(x) = 1$ if consumer x buys product j and $\omega_j(x) = 0$ if consumer x does not buy product x. Consumer x will only be able to choose to buy product j if $b_j(x) > b_{1-j}$.

There is a continuum of consumers with taste characteristics or locations distributed in a uniform density normalized to one along the real number line. Each consumer $x \in \mathbb{R}$ chooses either to buy: (a) the product revealed to him through advertising or (b) some outside option to maximize his utility $u(x)$. Consumer x is only shown the product $j = 0, 1$ where $b_j(x) > b_{1-j}(x)$. If consumer x buys good j, then he gets a utility of $u_j(x) \equiv R - t|x_j - x| - p_j$, and if consumer x takes the outside option, then he gets a utility of 0. Therefore, $\omega_j(x) = 1\{u_j > 0\}1\{b_j(x) > b_{1-j}(x)\}$. Each firm j's profit from consumer x is

[4] I ignore the degenerate equilibria, such as when one firm bids zero and the other firm bids high.

[5] In Appendix A, I extend this to a market equilibrium by introducing endogenous product prices and many firms.

given by equation (1).

$$\Pi_j(x) = [p_j * 1\{u_j > 0\} - b_{1-j}(x)]1\{b_j(x) > b_{1-j}(x)\} \tag{1}$$

2.2 Firm Bidding Strategy for Individual Consumers

Figure 1 shows firm 0's pricing decision. Firm 0 can potentially sell to consumers located between $x_0 - \frac{R}{t}$ and $x_0 + \frac{R}{t}$ from its location of x_0. Consumer's valuations for firm 0's product is the dotted line $R - t|x_0 - x|$. A consumer located at x_0 would be willing to pay up to R for the product, while consumers located at $x_0 - \frac{R}{t}$ and $x_0 + \frac{R}{t}$ would be willing to pay up to 0 for the product.

[Figure 1 about here.[6]]

If firm 0 sets a price of p_0, then consumers between $x_0 - \frac{R-p_0}{t}$ and $x_0 + \frac{R-p_0}{t}$ would be willing to buy its product if they see its ad. Consequentially, firm 0's value for consumers and by the same reasoning firm 1's value for consumers is given by equation (2). Because firms bid for advertising in a second-price auction, one bidding equilibrium is that all firms bid their value for winning each auction. In this paper, I will only consider this case.[7] Therefore, firms bid according to equation (2).

$$b_j(x) = \begin{cases} 0 & \text{if } x < x_j - \frac{R-p_j}{t} \\ p_j & \text{if } x \in [x_j - \frac{R-p_j}{t}, x_j + \frac{R-p_j}{t}] \\ 0 & \text{if } x > x_j + \frac{R-p_j}{t} \end{cases} \tag{2}$$

Figure 2 shows an example of an auction for targeted advertisements between firm 0 and firm 1. Firm 0 bids p_0 for consumers between $x_0 - \frac{R-p_0}{t}$ and $x_0 + \frac{R-p_0}{t}$ and 0 otherwise. Additionally firm 1 bids p_1 for consumers between $x_1 - \frac{R-p_1}{t}$ and $x_1 + \frac{R-p_1}{t}$ and 0 otherwise.

[6] See page 46 for figures.

[7] I ignore degenerate equilibria where one firm bids zero and the other bids more than either would be willing to pay to advertise to a consumer.

[Figure 2 about here.]

Therefore, for consumers between $x_0 - \frac{R-p_0}{t}$ and $x_1 - \frac{R-p_1}{t}$: firm 0 wins the auction and pays firm 1's bid of 0. Firm 0 makes p_0 from each of these consumers. For consumers between $x_1 - \frac{R-p_1}{t}$ and $x_0 + \frac{R-p_0}{t}$: firm 0 wins the auction and pays firm 1's bid of p_1. Firm 0 makes $p_0 - p_1$ from each of these consumers. For consumers between $x_0 + \frac{R-p_0}{t}$ and $x_1 + \frac{R-p_1}{t}$: firm 1 wins the auction and pays firm 0's bid of p_0. Firm 1 makes p_1 from each of these consumers. It follows that firm 0's profit is $\Pi_0 = p_0[(x_1 - \frac{R-p_1}{t}) - (x_0 - \frac{R-p_0}{t})] + (p_0 - p_1)[(x_0 + \frac{R-p_0}{t}) - (x_1 - \frac{R-p_1}{t})]$, firm 1's profit is $\Pi_1 = p_1[(x_1 + \frac{R-p_1}{t}) - (x_0 + \frac{R-p_0}{t})]$, and the advertising revenue is $\Pi_A = p_1[(x_0 + \frac{R-p_0}{t}) - (x_1 - \frac{R-p_1}{t})]$.

Figure 3 demonstrates the difference between niche and general products. Expensive luxury cars like a 2011 Lexus LS 460, where 9,568 units were sold in the US with a manufacturer's suggested retail price of $73,000, are niche products. In contrast, cheaper mass-produced cars like a 2011 Toyota Corolla, where 240,259 units were sold in the US with a msrp of $16,230, are general products.[8] Niche products often are more expensive, because of less competition, better ability to price discriminate through selling many niche products instead of one general product, and higher marginal cost of production from not taking as much advantage from an economy of scale.

[Figure 3 about here.]

Yet in a full model, niche-ness is a function of consumer preferences, and product price is a result of niche-ness and other market characteristics; in the real world not all niche products are expensive.

2.3 Webpages

In this section, I extend the model presented in subsection 2.1 to include webpages. I do this by splitting the set of tastes or locations x into 'webpage' intervals. Each interval

[8] Figures from Toyota press releases available at http://pressroom.toyota.com.

represents all the consumers visiting one webpage, which is assumed to be an exogenous process. Each firm j's advertising bidding function b_j is restricted to be constant for each interval. Therefore, firms can only target consumers based on which webpage they visit.

Figure 4 shows the case of one advertiser. The consumers not visiting the webpage, or equivilantly not visiting the webpage interval, cannot buy the product, because they would never be informed about the product through the advertising on the webpage. The advertiser only values consumers who see the ad and would buy the product. He values advertising to those consumers p_0. He bids a per consumer bid b_0 equivalent to his average profit per consumer who visits the webpage.

[Figure 4 about here.]

Although this is a pay-per-impression (PPI) auction, where the bidding is based on the number of consumers who see the ad, there are a number of other equivalent auctions. This auction is equivalent to each firm j making one bid B_j for all advertising on the webpage, where the highest bidder pays the second highest bid. It is also equivalent to a simple pay-per-click (PPC) or pay-per-action (PPA) auction, where firms pay per the number of consumers who buys its product or clicks on its ad.[9] Here, if a mass q_j of the consumers on the website would buy firm js product if it saw firm js ad, and if firm j bids $\tilde{b}_j$, then the winning bidder would be the firm with the highest $\tilde{b}_j q_j$ and would pay $\tilde{b}_{1-j} q_{1-j}/q_j$ per consumer who buys its product or clicks on the ad.[10]

Figure 5, demonstrates how a more narrowly-focused webpage would deliver a smaller segment of consumers to advertisers. A generally-focused webpage, like 'Ford', would deliver a larger, more heterogeneous interval of consumers to advertisers. A more narrowly-focused webpage, like 'Ford Trucks', would deliver a smaller, less heterogeneous interval of consumers to advertisers. An even more narrowly-focused webpage, like 'Ford Truck Parts', would deliver an even smaller, even less heterogeneous interval of consumers to

[9] Assuming that these are equivalent and we are not playing a search game.

[10] In a more complicated model of an auction for online advertising this result breaks down (Agarwal et al., 2009, see for example:).

advertisers. Note that webpage narrow-ness is not related to product niche-ness. They can be different segments along the same number line.

[Figure 5 about here.]

Consider the case of the three webpages depicted in Figure 6. Webpage (a) is the most generally-focused webpage, webpage (b) is more narrowly-focused than webpage (a), and webpage (c) is the most narrowly-focused. Firm 0 is more niche than firm 1 because it has a smaller interval of potential consumers.

[Figure 6 about here.]

For each firm $j \in \{0, 1\}$, firm j's bid for advertising to all consumers in webpage interval (a) is $B_j + B_s$. Therefore, if $B_j > B_{1-j}$, then firm j will win the auction for advertising to all the consumers in the webpage interval. Consequentally, firm 0 wins the auctions for advertising on webpages (a) and (c), and firm 1 wins the auctions for advertising on webpage (b). General webpages, like webpage (a), deliver a large variety of consumers. Firm 0 wins the auction for webpage (a), because it values a large variety of consumers. As a webpage becomes more niche, like going from webpage (a) to webpage (b), the webpage will attracts a more homogeneous set of consumers. A more niche firm will then be able to make more profit from each of these consumers through its higher prices. Firm 1 wins the auction for webpage (b), because it values each of its niche group of consumers more than firm 0. As a webpage becomes even more niche, the chance of it matching with a niche product decreases. Firm 0 wins the auction for webpage (c) because its broadness allowed it to value consumers that firm 1 did not.

3 Data & Experimental Design

In this section, I describe the online field experiment that I ran to test whether an ad for a product targeted toward a smaller market segment selling a niche product or an ad for a

product targeted toward a larger market segment selling a general product would win the auction for different types of webpages that were broadly-focused and narrowly-focused. I create webpages that each contain: (1) a unique content and (2) a space for one Google text advertisement. For each webpage, Google observes the title and auctions advertising on that webpage based on the perceived topic of the webpage. I collect my dataset by having my Firefox browser randomly cycle through my webpages, and record the ads on each webpage.

3.1 Creation of Webpages

Each of my webpages contains content I chose and one ad determined by the advertisers in an auction. For example, in Figure 7, I posted the content 'Bentley Convertible.'[11] Google auctioned the ad space in an online auction. The winning ad (i.e. the only ad I observed) was "New 200 Convertible: Build & Price Your New Chrysler® 200 Convertible. Find a Dealer Now!"

[Figure 7 about here.]

I signed up for having online contextual Google advertising for my webpages using Google's online advertising program: 'Google Ad Sense'.[12] This allowed me to create frames on my webpages where Google can place targeted ads.[13] By choosing the size of the frame of the ad, I chose the number of ads that I could see on my webpage. I chose

[11] I posted the content of my webpages in two places: (1) in the body of the webpage and (2) in the title of the webpage (the string of text between the Firefox symbol and "Mozil..."). For the purposes of this paper, I always set the content in those two places as equal strings of text, and I refer to that string as the content of the webpage. Future marketing research could set them as unequal strings of text to see the relative weight Google places on the title versus the body of the webpage.

[12] I read the advertising contract that I made with Google to make sure I was not violating it.

[13] The process of observing what content is on a webpage is not instantaneous. Google has to first observe the webpage, through an initial visit to the webpage. It then takes time, usually 15 minutes to a few hours, for Google to change the advertising from a general advertisement to a targeted advertisement. People in the computer science field refer to this process as *indexing* the webpage. I allowed Google a full day before gathering any of my data regarding the ads; Google had much more time than it usually takes to index each of the webpages.

ads of 125 pixels by 125 pixels, which is only big enough for one text ad of no more than 100 characters (including spaces). I also restricted Google to only allow text ads on my webpages.[14]

I posted my webpages under a domain name (or URL) that I acquired through GoDaddy in 2010. I could not completely randomize my choice of my domain name because Google often chooses to target based on the domain name and even a random name might contain a phrase that advertisers target based on. Therefore, I chose a six character domain name that did not receive any search results on Google.[15] I have left the domain name out of the paper, so that I may continue to use the domain name for future experiments.[16]

I set up my webpages so that I would receive as little outside traffic as possible. For instance, I created a false main page that does not link to any of the webpages that I use in my experiment. This way, any surfers, Google employees, or web crawlers would stumble on my false main page when then enter in my domain and not the webpages I use in my experiment. [17] Throughout my experiment, Google never linked to any of my webpages, and I received no clicks on any of my ads.[18]

3.2 Choice of Webpage Content

One of the big advantages of creating my own webpages, is I can control what content is on them. For example, if I wanted to compare the advertising on a 'sports car' webpage with the advertising on a 'red sports car' webpage, I could create two webpages: one with the content 'sports car' and another with the content 'red sports car' instead of having to find two similar webpages or magazines that only vary in the word 'red'. I could even

[14] I could have allowed picture ads or video ads. I chose not to so that my sample would be only text ads.

[15] In addition, I chose the domain name such that any contiguous string of three or more characters in the name did not receive any search results on Google.

[16] It is available on request.

[17] This false main page was also necessary for my application to have advertising. A Google employee manually looks at the main page of anyone applying to the 'Google Ad Sense' program.

[18] In the three years, I have only received four clicks on any of my ads. Given my hundreds of thousands of observations, I don't think this had any effect on how ads were targeted on my webpages.

test the ad results on topics not already covered by previously existing webpages. For example, there may not exist a 'red sports car' magazine, but I can create a 'red sports car' webpage.

Artificially constructing webpages in this manner has the disadvantage of not being similar to previously-existing content on the Internet. Because of this disadvantage, I chose to use the text content[19] from previously existing webpages on the Internet as the text content of my webpages.[20]

I chose to use content from websites that were currently being auctioned on Flippa.[21] Flippa (flippa.com) is the largest online marketplace dedicated to buying and selling websites. For instance, nearly 26,000 websites were sold on Flippa in 2011, for a value of almost $31 million.[22] The advantage of using websites currently auctioned on Flippa is that the seller provides data on the highest bid (i.e. the price for the website), age of the website, ad revenue, website traffic, website ranking, and many more variables. I did not use previously-auctioned websites despite that data being available, because the buyer could have changed the website's characteristics after the auction.

It is reasonable to assume that any information provided about a website on Flippa is accurate because (1) most of these statistics are verified by third party sources such as Google and (2) sellers sign a legal contract that their information is accurate. Despite this, there is plenty of missing data, because sellers did not have to share any information on a website. To account for these issues and avoid an issue of selection bias, I only used the final winning bid for the website, which is available for every website. I used it as a proxy for the advertising revenue of the website.

I chose to use the sixty-nine websites that met all of these criteria: (1) the website was

[19] I removed the HTML code and only used the text content.

[20] In order to avoid copyright issues, I randomized the order of the words. Randomizing the order of the words has no affect on the algorithms I use in this paper. It may affect the output of the algorithms used by Google. I leave this question to future marketing research.

[21] These websites were being auctioned as of May 23rd 2013.

[22] For more information on Flippa see https://flippa.com/about.

being currently auctioned[23] on Flippa on May 23rd 2013,[24] (2) the website was identified as an automobile website;[25] (3) the URL was not hidden by the seller;[26] and (4) the website had text content;[27].

I programed a webcrawler to gather the text off of each of the webpages. For each website, the webcrawler started by downloading the main page where the URL was provided by the Flippa data. Then it gathered each webpage linked to from the main page within the website domain.[28] It next gathered each webpage linked to those pages and iterated, with a maximum of five webpages per website. This was a self-imposed limit because some websites had hundreds of webpages, and some had only a few webpages.[29]

For each webpage: I next took the text content off of the webpage;[30] I randomized the order of the words;[31] and I used the text as the content of a new webpage as described in section 3.1.

3.3 Data Collection

I then ran an automated program to observe advertisements on my webpages. My data collection program:[32] (1) randomly brings up one of my webpages in the firefox browser;

[23] through a first-price sealed bid auction with the possibility of a reserve price and the possibility of a buy-it-now price.

[24] 65 of the 1,334 websites that were currently for sale were not being auctioned. Instead. the seller set it as a 'private sale', which means that buyers would make offers and the seller would choose which offer to accept. I did not use these websites because the final price would not be published on Flippa.

[25] Either by self reported categories or a search for the word 'automobile'. This way the websites were all in one industry, the automobile industry.

[26] In Flippa, sellers have the option of hiding their URL and only revealing statistics on the site they are selling. The idea behind this is that some sellers may not want their users or some other party knowing they are selling their website. It would have been impossible for me to use content of their sites because I did not know which site was theirs. Of the 1,334 websites that were currently being auctioned on May 23rd 2013, only 14 had their URL hidden.

[27] Some sellers choose to sell domain names without websites that presumably are premium url names.

[28] A website's domain is all the stuff before the '.com', '.net', etc.

[29] I did this to prevent myself from creating too many webpages and biasing my observations too much toward those with many webpages.

[30] I removed any HTML code through sophisticated use of regular expressions.

[31] I did this to avoid any possible copyright infringement on any webpage.

[32] I thank John, Thomas, and Anna Bruestle for their programming help on writing my data collection program. The program collects data from the firefox browser, not the source code, because Google pro-

(2) it grabs the webpage content, the text ad, the date, and the time off the webpage; [33] (3) it randomly jumps to another of my webpages in Firefox. Then it iterating back on (2) to grab the information off the new webpage.

Although one iteration can take a fraction of a second, I purposely slowed it down to 20 iterations per minute to keep the traffic on my webpages low. Although technically this experiment does not violate the advertising contract I made with Google,[34] I did not want to gain special attention from Google. Google tracked how much traffic each of my webpages gets per day, including my automated program. If I had received too much traffic, then I would likely have gained the attention of a Google employee who might choose to shut down advertising on my webpages. By keeping my traffic low and avoiding clicking on my ads, I am confident that my experiment escaped notice or at least Google did not choose to do anything that would impact my study.

3.4 Data Description

From May 27th to September 21st 2013, I collected 893,614 observations like the sample observations I show in Table 8.

[Table 8 about here.]

For every observation, I collected: the time and date, webpage content, ad URL, and ad textual content. Different observations of the same webpage sometime gave me different ads, as shown by my two observations of my "configuration diesels ... " webpage in my sample data. Therefore, I did not observe a single ad for each webpage, instead, I observed a single ad for each observation. The same ad is often observed on different

grammers made it difficult, if not impossible, to get the ad directly from the source code. It was written in Javascript embedded into the webpages in a way such that Google would not be able to see the program.

[33] The program gathers data by printing the output to a text file, because Google programmers have made it impossible to gather data through the source code of my webpages.

[34] I did not click on any of the ads on my own webpages.

webpages, as shown by my two observations of an ad for the Trenton used car sale in my sample data.

Each ad contains an average of 13.939 words, for a total of 12,456,083 words. When I removed repeated ads, each ad contains an average of 13.685 words per ad, for a total of 218,555 words. Each of my 138 unique webpage contains 202.203 words for a total of 27,904 words. I often observed the same word repeated in different ads and webpages; I observed only 20,014 unique words.

4 Topic Modeling

Intuitively, a Lincoln Blackwood, which is Ford's 2002-03 all-black luxury pick-up truck which sold only 3,356 units, is a niche product. In contrast, a Toyota Corolla, which is the perennial best selling car in the world, is not. Therefore, an ad for Lincoln Blackwoods would be more niche than an ad for Toyota Corollas. And a webpage on Lincoln Blackwoods would be more narrowly-focused than a webpage on Toyota Corollas. The methodological problem is finding some scientific and systematic way of measuring how niche an ad is and how narrow a webpage is. This is not trivial, because the measure needs to be exogenous. Therefore, I cannot use number of units sold, price of the product, or price of the ad as proxy variables for niche-ness and narrowness, because these are all endogenous measures.

While an ideal experiment might allow me to pick both the possible ads and the possible webpages, in this experiment, I could not choose the possible ads; instead, I observed real text ads created and bid on by real advertisers. Therefore, I could not slightly vary the text in ads to observe different levels of targeting. I could not add the word 'red' or the word 'truck' to the text of an ad and see how it changed the ad's performance. Instead, I observed the occurrences of different ads. Some of these contain the word 'red', some of these contain the word 'truck', and some of these did not contain either word.

While I could choose to vary my webpages by adding the word 'red' or the word

'truck', there is no way of knowing how that changes the narrowness of a webpage, without some way of measuring the narrowness or the niche-ness of a word derived from how words are used in real ads and webpages. I consequentially chose to use the words from real webpages (randomizing to avoid copyright issues). Because of this, I was also restricted by the words on the webpages I observed.

Furthermore, I could not simply choose a set of keywords in an ad to count with the idea that an ad is more specific if it has more keywords. These measures face, the same problem that was faced in previous studies in economics using keywords in newspapers to identify political bias (see for example: Agirgas, 2011; Gentzkow and Shapiro, 2006; Larcinese et al., 2011). I would be choosing and justifying the set of keywords to count in a non-empirical way. I would be making a judgement call. While I may have been able to justify a set of keywords heuristically, I could not ignore the possibility that there was another set of heuristically justifiable keywords that I had not considered.

In previous versions of this experiment, I tried various heuristic measurements of ad niche-ness and webpage narrowness including: (1) the number of Google search results of a word where fewer results meant a niche word, (2) the number of nouns in a ad or webpage where more nouns meant the ad or webpage was more specific, and (3) the number of automobile manufacturer and make names where more of these keywords mentioned meant the ad or webpage was more specific. All these test produced weak and inconclusive results.

I construct a measure for ad niche-ness using a *topic modeling* algorithm. I estimate the niche-ness of each word in each ad through examining latent patterns between clusters of words in my ads. Topic modeling algorithms are probabilistic algorithms for uncovering the underling structure of a set of documents using hierarchical Bayesian analysis of the original texts. They are most often used to categorize documents based on observed patterns of words in the texts.

In the past decade, most of the development of topic modeling has been from adaptations and applications of Blei et al.'s (2003a) Latent Dirichlet Allocation (LDA). This in-

cludes analysis of scientific abstracts (Griffiths and Steyvers, 2004) and newspaper archives to be generated from multiple topic; in this study, it allowed the topics of documents to be estimated without having to create a new exogenous variable for each document. It estimates the latent random drawn of topics and identifies the underlying latent parameters of its model quickly. Many of the algorithms based on LDA only take a few hours to run on a set of tens of thousands of documents. For a good overview of the important topic models and the importance of LDA, see Blei and Lafferty (2009).

In 4.1, I will first show you the basic topics estimated by the LDA model. Then, I will present the LDA model (4.2), show that the underlying parameters of the model are identified (4.3) by variation in the data, and discuss the algorithms used to estimate the LDA model (4.4). I present the results from running an LDA algorithm on my model first because I seek to illustrate what the model does before we get into the details of how it works.

In 4.5, I present Blei et al.'s (2003b) hierarchical extension to the LDA model, which I use to construct a measure of niche-ness/narrowness. I find strong evidence for a non-monotonic relationship: The moderately narrow webpages have the less niche ads.

4.1 Latent Dirichlet Allocation Estimation of My Data

Table 9 shows five categories identified from the 15,970 unique ads and 138 unique webpages using Blei et al.'s (2003a) mean field variational inference algorithm for the LDA model.[35] Each column shows the estimated top thirty words for each category.[36]

[35] I describe this algorithm in 4.4.

[36] The most common word in one topic could be common in another topic, so generally the most frequent word in a topic is not considered the top word in a topic. Top words are those with the highest term-score (from: Blei and Lafferty, 2009, see equation (3)), which was inspired by the Term Frequency and Inverse Document Frequency (TFIDF) score of vocabulary terms used in Baeza-Yates and Ribeiro-Neto (1999).

$$term\_score(k,v) = \hat{\beta}_{k,v} log(\frac{\hat{\beta}_{k,v}}{(\Pi_{j=1}^{K} \hat{\beta}_{j,v})^{1/K}}) \tag{3}$$

Here, $\beta_{k,v}$ is the probability of observing vocabulary word v in topic k and $\hat{\beta}_{k,v}$ is its estimate.

Based on these estimations of word clusters, the econometrician chooses names for each topic. These names should be interpreted as how the econometrician interprets the data.[37] For example, Topic 1 seems to be composed of many words under the 'Computing & Servers' topic (ex: 'cloud', 'hosting', 'ftp', etc.). Additionally, Topic 2 seems to be composed of many words under the 'Tires & Cars' topic (ex: 'tire', 'michelin', 'wheels', etc.).

[Table 9 about here.]

Figure 10 shows the topics for sample documents from my data. In LDA, each word is a latent draw from a single topic, and different words in the same document may be drawn from different topics. LDA is a *mixture model*. It allows for documents to be drawn from multiple topics. For example, the bottom document in figure 10 is estimated to have been drawn from topics 1 through 4.

[Figure 10 about here.]

4.2 Latent Dirichlet Allocation Model

In this section, I present the LDA model from Blei et al. (2003a) to introduce the basic topic model. In 4.5, I will present and use an extension of this model that allows for hierarchical topics to estimate the niche-ness of the words in my ads.

In LDA, I assume that all the words in D documents, which is the set of all unique webpages and unique ads, drawn from a set of V vocabulary words through the following latent process:

For each document $d = 1, ..., D$, the number N_d of words in the document d are drawn from some random distribution. The assumption of what random distribution is not critical to anything that follows because I am modeling the choice of words, not the choice

[37] Sometimes the names are chosen as the top word for a topic.

of the number of words. It does not have to be independent; it can be correlated with the other data generating variables. In the case of my ads, Google limits the number of characters an advertiser uses to one hundred. In general, it is treated as an ancillary variable and is treated as exogenously given.

Next, a K-dimensional random vector $\vec{\theta}_d$ is drawn from a Dirichlet Distribution with a K-dimensional parameter $\vec{\alpha}$, where K is an exogenously given number of topics. Each document is drawn from each topic in different proportions. $\vec{\theta}_d$ is the vector of these proportions for document d. The k^{th} element $\theta_{d,k}$ of vector $\vec{\theta}_d$ will be the probability any given word in document d is drawn from topic k. The Dirichlet distribution draws random variables $\vec{\theta}_d$ on the $(K-1)-$simplex ($\sum_{k=1}^{K} \theta_d k = 1$) and has a probability density function given by equation (4).[38]

$$p(\vec{\theta}_d|\vec{\alpha}) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\Pi_{k=1}^{K} \Gamma(\alpha_k)} \Pi_{k=1}^{K} \theta_{d,k}^{(\alpha_k - 1)} \tag{4}$$

Then, for each word $n = 1, ..., N_d$ in document d, a random topic $z_n \in 1, ..., K$ is drawn from the multinomial distribution with a parameter of $\vec{\theta}_d$. The probability of choosing topic k is $\theta_{d,k}$. The draws of topics are technically not independent because they depend on the document specific parameter $\vec{\theta}_d$. They are essentially *conditionally independent and identically distributed*, where the conditioning refers to the document specific parameter $\vec{\theta}_d$. Therefore, in LDA, it is assumed that the order of the words in a document does not matter.[39]

Then for each word $n = 1, ..., N_d$ in document d, a random word $w_n \in 1, ..., V$ is drawn from the multinomial distribution with a parameter β_{z_n} conditioning on the topic z_n of word n. The probability of choosing vocabulary word v is the parameter $\beta_{z_n,v}$.

An example of this process is shown in Figure 11. There are two bags of words. Each

[38] It is standard practice to restrict all the αs to be equivalent, which is equivalent to restricting all topics to be about the same size. This makes makes parameter estimation much quicker. I do not do this here because the HLDA model extends the more general Dirichlet distribution.

[39] This assumption is later relaxed in dynamic topic models (DTM) (Blei and Lafferty, 2006, see:).

bag of words is a different topic ($K = 2$). Topic 1 is 'Trucks'. A random word from a document about 'Trucks' has a 25% chance of being the word 'pickup', a 20% chance of being the word 'truck', and so forth. Topic 2 is 'SUVs'. A random word from a document about 'SUVs' has a 25% chance of being the word 'suv', a 20% chance of being the word 'toyota', and so forth.

[Figure 11 about here.]

For each document $d = 1, 2$, and 3, the number of words is drawn from some random process. There are $N_1 = 11$ words in document 1, $N_2 = 8$ words in document 2, and $N_3 = 15$ words in document 3. Then, for each document d, a vector $\vec{\theta}$ is drawn, representing the probability from drawing from each bag of words. For words in document 1, the probability from drawing words from the first bag (topic 1) is 100%, so $\theta_1 = 100\%, \theta_2 = 0\%$. For words in document 2, the probability from drawing words from the first bag (topic 1) is 50% and from the second bag (topic 2) is 50%, so $\theta_1 == 50\%, \theta_2 = 50\%$. Likewise, $\theta_1 = 0\%, \theta_2 = 100\%$.

Words are then selected for each document d in the following way: For each word $n = 1, ..., N_d$, a random bag of words (or topic) is drawn. The probability of drawing the bag of words k (topic k) is $\theta_{d,k}$. Therefore all the words in document 1 are drawn from the first bag of words. All the words in document 3 are drawn from the second bag of words. Each word in document 2 then has a 50% chance of being drawn from the first bag of words and a 50% chance of being drawn from the second bag of words.

4.3 Identification

The advantage of the LDA model over its predecessor Probabilistic Latent Semantic Indexing (pLSI) model (Hofmann, 1999) is that the LDA can easily be identified. In this section, I will explain why we have enough observations to identify the parameters in the LDA model, and I will explain why we observe enough variation in the data to confidently

identify LDA. I do this by first discussing identification in the pLSI model, which I use to explain the identification in the LDA model. This same discussion applies to the HLDA model, which is a restricted version of the LDA model.

The pLSI model (Hofmann, 1999) is the first mixture model in topic modeling; it is the first topic sorting model that allows for documents to be drawn from multiple topics. The basic probability of a realization of a document given by the pLSI model is shown in equation (5).

$$p(\vec{w}|d,\beta) = \Pi_{n=1}^{N_d}\Big(\sum_{z_n}\beta_{z_n,w_n}\theta_{d,z_n}\Big) \tag{5}$$
$$\beta_{k,v} = p(w = v|z = k)$$
$$\theta_{d,k} = p(z = k|d)$$

Here, each document d is a realization of a N_d-vector of words $\vec{w}$. Each word is selected by first drawing a random topic from the document specific distribution of topics where $\theta_{d,k}$ is the document specific probability of drawing topic k. Then a vocabulary word is drawn from the topic specific distribution of words where $\beta_{k,v}$ is the topic k specific probability of drawing vocabulary word v. Therefore the pLSI model has $K(D + V)$ underlying parameters: K parameters for each document (each $\theta_{d,k}$) and K parameters for each word (each $\beta_{k,v}$).

The LDA model (Blei et al., 2003a) simplifies the pLSI model to allow for relatively fewer parameters to identify. The probability of a realization of a document given by the LDA model is shown in equation (6).

$$p(\vec{w}|\vec{\alpha},\beta) = \int p(\vec{\theta}_d|\vec{\alpha})\Pi_{n=1}^{N_d}\Big(\sum_{z_n}\beta_{z_n,w_n}\theta_{d,z_n}\Big)d\theta \tag{6}$$

Each probability $\theta_{d,k}$ of a topic k in document d is now a random realization from the same Dirichlet distribution. Therefore the basic LDA model has $K(1 + V)$ underlying parameters: the K-parameters in the vector $\vec{\alpha}$ that control the Dirichlet distribution and K parameters for each word. Adding another document to the dataset adds to the number of underlying parameters in the pLSI model by K, but does not add to the number of underlying parameters in the LDA model unless a new word is drawn.

Increasing the number of documents collected increases the number of vocabulary words. An additional document should be expected to add new words to the vocabulary list at a decreasing rate. This poses two identification problems: (1) the maximum likelihood estimates of the basic model would say that this is a zero probability event, and (2) there will always be a set of rare words with few observations.

Blei et al. (2003a) solved both of these problems by applying a form of smoothing to the $K \times V$ parameters β.[40] They treat each vector $\vec{\beta_k}$ (the set of probabilities of seeing each word in each topic k) as a random draw from a symmetric Dirichlet distribution with a latent parameter η.[41] This produces a random vector on the $V - 1$-simplex. If a new word is added to the vocabulary set, then this becomes a random draw on the V-simplex. Therefore, every word observed has a positive probability of being drawn in each topic (solving problem 2), and there is a positive probability of drawing a new word when extrapolating to new documents (solving problem 1). Consequentially, the LDA model has $K + 1$ parameters that need to be identified: $\vec{\alpha}$ and η. Everything else ($\theta, \beta, F, \ldots$) are all random variables that we can estimate given identifying $\vec{\alpha}$ and η.

If α_k is relatively smaller than the other αs, then Topic k occurs relatively less frequently than the other topics. If α_k is relatively larger than the other αs, then Topic k occurs relatively more frequently than the other topics. Consequentially, the relative size of the αs is identified by the frequency of occurrences of documents with that topic, or in other words, by how common the cluster of words appears in my data.

[40] Unfortunately, simple Laplace smoothing is no longer justified as a maximum a posteriori method for this type of problem.

[41] $\vec{\eta} =< \eta, \eta, \eta, \eta, \ldots >$.

Figure 12 shows how $\bar{\alpha} = \sum_{k=1}^{K} \alpha_k$ is identified. $\bar{\alpha}$ controls the mixtures of topics in documents. A large $\bar{\alpha}$ occurs when there is a large overlap in the clusters of words used for topics. A small $\bar{\alpha}$ occurs when there is a small overlap in the clusters of words used for topics. A large $\bar{\alpha}$ means the topics are very similar and the words in each document tends to be drawn from a a more even spread of topics. A small $\bar{\alpha}$ means the topics are very disjointed and documents are usually drawn from only one topic.

[Figure 12 about here.]

η is identified by looking at the distribution of words in the topics. If the words in each topic are drawn mostly from an even spread of probabilities across a large cluster of words, then η is relatively large. If the words in each topic are drawn mostly from a small cluster of a few words, then η is relatively small.

4.4 Overview of Estimation Techniques

The key to identifying the LDA model is through the likelihood function given by equation (6). Unfortunately, equation (6) is too intractable to estimate reliably using maximum likelihood because of the coupling between θ and β (see: Dickey, 1983). Several approximation techniques have been developed. The most commonly accepted are: mean field variational inference (Blei et al., 2003a), Gibbs sampling (Griffiths and Steyvers, 2004), expected propagation (Minka and Lafferty, 2002), and collapsed variational inference (Teh et al., 2006a). In this section, I will describe the two most common techniques to illustrate the general idea of how they work. I refer to the authors of these techniques for the details.

Blei et al. (2003a) developed the first algorithm as a mean field variational inference technique, which is what I used to identify the topics used in Tables 9 and 10. This basic idea of this technique is to: (1) use Jensen's inequality to find an adjustable lower bound on the log of the likelihood given in equation (6). The estimates for the document and word-specific parameters ($\vec{z}$ and $\vec{\theta}$) are then chosen that produce the tightest possible

lower bound. Then (2) use these estimates to find the best estimates for the document generating parameters (β, $\vec{\alpha}$ and η), then iterate back on (1). The error introduced by Jensen's inequality will converge to zero as the parameter estimates converge.

The most commonly accepted algorithm (mostly from its speed) is Griffiths and Steyvers's (2004) Gibbs sampling technique. I will use an adaptation of this technique to identify the HLDA extension to LDA in 4.5. Gibbs sampling is a form of Markov Chain Monte Carlo (MCMC). The basic idea of this technique is that all of the parameters can be estimated from the realizations of all the words topic assignments z. Therefore, posterior estimates of the set of probabilities that a word is in each topic can be calculated from the topic assignments of the other words. The algorithm (1) assigns each word an arbitrary topic,[42] (2) calculates the set of probabilities that a word is in each topic from the current topic assignments of the other words, and (3) randomly draws a topic for each word from its distribution. It then iterates back on (2).

4.5 Hierarchical Latent Dirichlet Allocation

In this section, I restrict the LDA model to have a hierarchical topic structure (Blei et al., 2003b). I begin by defining the hierarchical structure of topics through the *nested Chinese restaurant process*. Then, I explain how this modifies the LDA model. In the next section, I will use these results to analyze the effect of webpage narrowness on ad niche-ness.

Imagine a process where M customers enter a Chinese restaurant. The first customer sits at the first table. Each additional customer m sits at a random table from the probabilities given in equation (7).

$$\begin{aligned} &p(\text{occupied table } k|\text{previous customers}) = \frac{m_k}{\gamma+m-1} \\ &p(\text{start a new table}|\text{previous customers}) = \frac{\gamma}{\gamma+m-1} \end{aligned} \tag{7}$$

Here, m_k is the number of previous customers sitting at table k, and γ is a parameter.

[42] This could be a guess or a random assignment.

Customers are more likely to sit at a table with more people because people are drawn to others. As more customers enter the restaurant, it is less likely that a customer will sit at a new table.

On each table, there is a flier with instructions to a new and different Chinese restaurant. All the customers at each table read their flier and decide to go to the new restaurant the next day. All the customers at the table start the process over again at the new restaurant the next day. In this way, each customer to draws a random path down a hierarchical tree of chinese restaurants.

This process is known as the *nested Chinese restaurant process*. In this same way, each document d is assumed to have drawn a random path down a hierarchical tree of topics. Once the path down the tree of topics is selected, then the words in the document are generated by an LDA process among the topics in its random path, as shown in figure 13.

[Figure 13 about here.]

This introduces two new parameters into the model: γ (which controls the probability of sitting at a new table) and the number of levels of the hierarchical tree. In general, HLDA algorithms do not estimate these parameters. These parameters control how many topics observed in the data. Optimizing the number of topics, usually produces many small topics (Blei et al., 2003a). To make the results meaningful, I limit the number of topics by setting $\gamma = .25$ and the number of levels to 3 (levels 0, 1, and 2). My results are robust for different γs around $\gamma = .25$, and $\gamma = .25$ is consistent with the machine learning literature (see 5.1 for details). I was unable to estimate more than three levels because of the small size of the documents. If the documents were longer, then I could estimate more levels.[43]

Because the number of topics is no longer fixed, the HLDA model treats the α (the parameters of the Dirichlet algorithm for drawing topic probabilities) as random draws

[43] The content on the webpages were long enough to estimate more levels. However, there were too few webpages for this to be meaningful.

from a GEM Distribution. This introduces two new variables: the mean and the scale of the GEM distribution. The GEM mean is the proportion of general words relative to niche words. The GEM scale controls how strictly documents should follow the general versus specific word proportions. I chose a GEM mean of .2 because it produced relatively few small topics (topics with only a few words) (see 5.1 for details). Additionally, I chose a GEM scale of 100% because I wanted to preserve a good balance of niche and general words. My results are robust for my choice of these parameters.[44]

This leaves two sets of variables for the algorithm to identify: $\vec{\eta}$ and α. As I mentioned in 4.3, Blei et al. (2003a) treats each vector $\vec{\beta_k}$ (the set of probabilities of seeing each word in each topic k) as a random draw from a symmetric Dirichlet distribution with a latent parameter η. In LDA, η is the same for all topics. In HLDA, we can relax this and let η be the same for all topics in the same level. Therefore, I have three parameters to identify η_0 (the η for the level 0 topic), η_1 (the η for the level 1 topics), and η_2 (the η for the level 2 topics), plus an additional parameter for each topic.

In running this experiment, I estimated the levels of the words by taking the mode of the estimated level from iterations 1,000, 2,000, 3,000, ..., and 10,000 of Blei et al. (2003b)'s Gibbs sampling algorithm for HLDA. The language modeling literature does it this way to reduce the error in estimating the levels of each word (see Blei et al. (2003a) for details).

5 Results

This section reports the results from my online field experiment. I start by explaining the process that I used to select reasonable input parameters for Blei et al.'s (2003b) Gibbs sampling algorithm for HLDA on my data set of unique webpages and ads (see 5.1). I next show the algorithm converges and the parameters are identified (see 5.2). Further, I analyze the distribution of topic levels of words in my webpages and ads (see 5.3), which I use to measure the niche-ness of words in my ads and narrowness in my webpages. I then

[44] I got similar results when I varied these parameters.

examine the top words from the topics, or equivalently the word clusters, estimated by the algorithm. I demonstrate that the algorithm estimates which words are niche and which words are general (see 5.4). I continue by regressing my measure of webpage narrowness on my measure of ad niche-ness and show that there is a non-monotonic relationship (see 5.5). I conclude this section by analyzing the residuals to make sure the estimates of my regressions are reasonable (see 5.6).

5.1 Input Parameter Determination

In this section I describe how I chose two of my variables: (1) γ = .25, which controls the probability of a document forming a new topic (see equation (7)), and (2) a GEM mean = .2, which controls the proportion of general words relative to niche words. The other input variables were chosen to be consistent with the literature.

Merely finding the optimum[45] number, size, and shape of the word clusters does not provide as meaningful a result. In Blei et al.'s (2003b) Gibbs sampling HLDA algorithm, there are several parameters the econometrician chooses to control the number, size, and shape of the word clusters that the algorithm identifies (see 4.5). It is likely that optimizing the number of topics, for example, would define each document as its own word cluster.

I find γ and the GEM mean by looking at the structure of the topic trees formed under different values of γ and the GEM mean. I then chose the topic tree that satisfies these three conditions: (1) it minimizes the number of small topics, (2) it minimizes the number of topics, and (3) it maximizes the number of level 1 and level 2 words. These conditions mean that my model fits the data better and therefore can produce more meaningful results.

For example, Table 14 illustrates how changing the γ, which controls the probability of a document forming a new topic (see equation (7)), affects the number, size, and shape of the topics estimated by Blei et al.'s (2003b) Gibbs sampling HLDA algorithm. Each

[45] Through finding the parameters that maximize likelihood.

column shows the topics estimated by the algorithm at the 10,000th and final iteration of the algorithm for three different γs: .2, .25, and .3.

[Table 14 about here.]

For example, for $\gamma = .25$, there is one level 0 topic, which is the general topic that is common to all 13,467 unique documents. It is estimated 122,177 words from these unique documents were drawn from this topic. There are ten level 1 topics, which are the topics more niche than the level 0 topic. These topics are estimated to have generated more than 41,065 of the words in my documents. And there are forty-four level 2 topics, which are the most niche topics; thirty-four of which generated an estimated twenty words or fewer.

A γ of .25 seems to fit better and have more meaningful results than a γ of .2 and .3, because (1) I get fewer small topics, (2) I get fewer topics, and (3) I get more level 1 and level 2 words.

Table 15 shows the number of topics in each level with a low word count by γ for *gamma* = .01, .05, .1, .15, .2, .25, .3, .35, and .4.

[Table 15 about here.]

Table 15 illustrates how $\gamma = .25$ produces fewer small topics: $\gamma = .25$ produces only one level one topic with fewer than fifty words and it produces fewer small level two topics than $\gamma = .2$ and $\gamma = .3$. Therefore, I feel justified in my choice of a $\gamma = .25$.

5.2 Convergence of Algorithm

In this section, I test the convergence of the Blei et al.'s (2003b) Gibbs sampling HLDA algorithm on my data. In language modeling, it is conventional to use perplexity instead of likelihood (see: Blei et al., 2003a; Rosen-Zvi et al., 2004). Perplexity is the predicted probability of being able to predict words in new unseen documents. Perplexity is monotonically decreasing in the likelihood and is equivalent to the inverse of the geometric

mean of the per-word likelihood. A lower perplexity indicates a higher likelihood, or equivalently better performance.

Figure 16 shows the perplexity for all 10,000 iterations of Blei et al.'s (2003b) Gibbs sampling HLDA algorithm conditional on the given parameters (γ and GEM mean) and the estimated parameters ($\vec{\eta}$, and $\vec{\alpha}$) being true. After the first thousand iterations, the perplexity has an average of about 2.176 million with a small standard deviation of 0.00758 million. Given that this is a Markov Chain Monte Carlo algorithm, and there would naturally be some variation in the estimation of the perplexity, this shows significant evidence that the model was identified.

[Figure 16 about here.]

Figure 17 (b) shows the perplexity conditional on only the $\vec{\eta}$ parameters being true. As I mentioned in 4.5, η_l is the symmetric Dirichlet parameter that determines the random draw of a probability $\beta_{k,v}$ of seeing a word v in a topic k from level l. This perplexity tends to be lower (indicating a higher likelihood) than the perplexity in Figure 17, because it is less constrained. After the first thousand iterations, this perplexity has an average of about 2.0444 million with a small standard deviation of 0.00796 million.

[Figure 17 about here.]

I could not do the same for the other identified parameters $\vec{\alpha}$, because the number of topics changes from iteration to iteration. Instead, I relied on the total perplexity discussed above and shown in Figure 16 to show that its estimates converge.

5.3 Distribution of Results

In this section, I examine the distribution of my HLDA measure. As I mention earlier, the levels of words were calculated by taking the mode of the estimated level from iterations 1,000, 2,000, 3,000, ..., and 10,000 of Blei et al.'s (2003b) Gibbs sampling algorithm for

HLDA. I then take an average of these levels for each document as my measure of ad niche-ness and as my measure of webpage narrowness.

Table 18 shows the sample frequency of the level of each word (level 0,1, or 2), first unconditionally and then conditionally on the level of another given word in the same ad or the same webpage.

[Table 18 about here.]

Table 18 first shows the sample frequencies across all 893,614 ads (Table 18a). Next, it shows the sample frequencies across all 13,467 unique ads (Table 18b), and it next shows the sample frequencies across all 138 unique webpages (Table 18c). The sample frequency $f(i|j)$ of the level of a word being i conditional on another given word in the same document (ad or webpage) being level j was calculated by

$$f(i|j) = \frac{\sum_{\mathrm{d}} \left(c_j^d * \frac{c_i^d - 1\{i=j\}}{c_0^d + c_1^d + c_2^d - 1} \right)}{\sum_{\mathrm{d}} c_j^d / d} \tag{8}$$

where c_i^j is the count of the number of words in document d that are of level i.

Across all observations (Table 18a), I estimate that 88.60% of the words in an ad were generated from the general level 0 topic, 10.13% from a level 1 topic, and 1.27% from a level 2 topic. If another given word is level 0, then I estimate that 89.82% of the words in an ad were generated from the general level 0 topic, 9.06% from a level 1 topic, and 1.15% from a level 2 topic. If another given word is level 1, then I estimate 83.23% of the words in an ad were generated from the general level 0 topic, 15.07% from a level 1 topic, and 1.71% from a level 2 topic. If another given word is level 2, then I estimate 83.28% of the words in an ad were generated from the general level 0 topic, 13.50% from a level 1 topic, and 3.26% from a level 2 topic.

Across unique ads (Table 18b), I estimate that 86.05% of the words in an ad were generated from the general level 0 topic, 12.11% from a level 1 topic, and 1.85% from a

level 2 topic. If another given word is level 0, then I estimate that 87.89% of the words in an ad were generated from the general level 0 topic, 10.54% from a level 1 topic, and 1.59% from a level 2 topic. If another given word is level 1, then I estimate 78.99% of the words in an ad were generated from the general level 0 topic, 18.41% from a level 1 topic, and 2.61% from a level 2 topic. If another given word is level 2, then I estimate 78.39% of the words in an ad were generated from the general level 0 topic, 17.25% from a level 1 topic, and 4.38% from a level 2 topic.

Across unique webpages (Table 18c), I estimate that 77.56% of the words in a webpage were generated from the general level 0 topic, 20.19% from a level 1 topic, and 2.25% from a level 2 topic. If another given word is level 0, then I estimate that 82.85% of the words in a webpage were generated from the general level 0 topic, 15.77% from a level 1 topic, and 1.38% from a level 2 topic. If another given word is level 1, then I estimate 56.75% of the words in a webpage were generated from the general level 0 topic, 41.31% from a level 1 topic, and 1.94% from a level 2 topic. If another given word is level 2, then I estimate 69.27% of the words in a webpage were generated from the general level 0 topic, 27.10% from a level 1 topic, and 3.62% from a level 2 topic.

If the conditional probabilities were $f(i|i) = 1$ and $f(i|\text{not } i) = 0$, then the average of the estimated levels of each word in a document or a webpage would either be 0, 1, or 2. Yet I observe a lot of mixing in the word level in my documents and webpages, so I observe many different average word levels between 0 and 2. If the conditional probabilities were such that $f(0|3) = f(3|0) = 0$, then the average estimated word level and number of words in a document would be sufficient to know how many words of each level are estimated to be in a document. In the case of my data, I am not losing much information about an ad or a webpage by looking at the average estimated word level, because I have (1) $f(i|i) > f(i|j \neq i)$ and (2) $f(1|2) > f(1|0)$ in all cases.

Figure 19 shows the cumulative distribution function of my HLDA measure of ad niche-ness for all 893,614 observations (the thick red line), for the 15,970 unique ads (the thinner gray line), and for the 77,507 unique ad-webpage combinations (the dashed blue

line). In addition, it shows the cumulative distribution function of my HLDA measure of webpage narrowness for the 138 unique webpages (the dotted green line). The CDF of my HLDA measure for all observations is weakly above the CDF of my HLDA measure for unique ads because the most common ads tend to be less niche. Because of this the CDF of my measure for unique ad-webpage combinations is sandwiched between the CDFs of my measure for all observations and unique ads (although very close to the CDF of my unique ads). In addition, the CDF of my HLDA measure for all webpages is weakly below the CDFs of my HLDA measure for my ads because webpages tend to be more narrow than ads are niche; webpages have more words, so they have a greater ability to become narrow than ads have at becoming niche.

[Figure 19 about here.]

In general, each of these CDFs looks like a smooth distribution that is cut off at zero.[46] For example, 27.08% of my unique ads and 32.52% of my observations have a measure of niche-ness of zero. This comes from all words in the ad or a webpage being estimated as being generated from the general level zero topic. Because the cut-off point at a HLDA measure of zero matters (more so for the ads than the webpages), I chose to analyze these data using a Tobit regression, which allows for a minimum and maximum observed dependant variable.

5.4 HLDA Estimates Niche-ness of Words

The purpose of this section is to determine if the HLDA algorithm gives good estimates for which words are general and which words are niche/narrow. I do this by looking at the top words from the common topics. As I mentioned in footnote 36, the most common word in one topic could be common in another topic, so generally the most frequent word in a topic is not considered the top word in a topic. Top words are those with the highest

[46] It is not perfectly smooth, because the number of words in an ad was limited by the 100 character max.

term-score (from: Blei and Lafferty, 2009, see equation (3)), which was inspired by the Term Frequency and Inverse Document Frequency (TFIDF) score of vocabulary terms used in Baeza-Yates and Ribeiro-Neto (1999). I only examine the most common topics because their large number of observations makes them estimated more accurately.

Figure 20 shows the top words from Topics Identified by Blei et al.'s (2003b) Gibbs sampling algorithm for HLDA. Each blue box represents a different topic identified by HLDA. It contains the words with the highest term-score for that topic. The algorithm identified fifty-five topics; Figure 20 shows the eight of the most common topics.

[Figure 20 about here.]

In level 0, there is one topic, which are those that are the common words. The top eight words from this topic are 'free', '&', 'the', 'your', 'to', 'for', 'a', and 'and'. These words are what I would expect to be common among automobile ads and webpages. They are also the words that I would describe as the most general, because they do little to differentiate automobile ads and webpages.

In level 1, there are ten topics, which are those that are the less common words than in level 0. The three topics shown are the only topics estimated to cover 2,862 or more words in my data (unique ads and webpages). These words would differentiate the ad more than those in level 0, so I would describe them as making the ad more niche. The words in these topics are more differentiated than those in level 0. For example, there are words like 'freight', 'scan', 'manuals', 'wheels', 'pm', 'may', and 'detroit'.

In level 2, there are forty-four topics, which are those that are the least common words. Figure 20 shows the four most common of these topics, which are the only topics that satisfy both: (1) have a parent topic of the three most common level 1 topics and (2) cover 170 or more words in my data. These words are the least common; they are more niche/narrow than level 0 and level 1 words. For example, there are words like 'influential', 'flowers', 'one-view', 'protocols', 'prohibited', 'frosty', 'jimmy', and 'e-bike'.

5.5 Webpage Narrowness vs. Ad Niche-ness

In this section, I compare webpage narrowness to ad niche-ness. I construct a measure of ad niche-ness and webpage narrowness by taking the average of the topic levels of the words that I estimated using HLDA.

Table 21 (b) shows my regressions of webpage narrowness, which is the average level of the topics of the words in an webpage, on ad niche-ness, which is the average level of the topics of the words in an ad.

[Table 21 about here.]

In Table 21 (b), I first ran the simple Tobit regression

$$y = \beta_0 + \beta_1 x + \epsilon \qquad \text{(R1 and R1a)}$$
$$\text{where } y \equiv \frac{c_1^a + 2 * c_2^a}{c_0^a + c_1^a + c_2^a} \in \{0, 2\}$$
$$x \equiv \frac{c_1^w + 2 * c_2^w}{c_0^w + c_1^w + c_2^w}$$

where the dependant variable x is the number of words in the webpage title, β_0 and β_1 are parameters to be estimated, ϵ is the error term where the conditional mean of ϵ given x is zero, and c_l^d is the count of the number of non-repeated words in the document(a for ad and w for webpage) that are estimated to be level $l = 0, 1, 2$. This would make my dependant variable y into the average of the estimated word level in the ad (my measure of ad niche-ness) and my independent variable x into the average of the estimated word level in the webpage (my measure of webpage narrowness). Note that y is bounded between 0 and 2 (thus the Tobit).[47]

Running this regression (R1) on all observations, I find that webpage narrowness increases my measure y of ad niche-ness by 0.012. This result is significant at the .1%

[47] x is also bounded, but this does not affect the regression.

level. Given that each ad contains an average of 13.939 words, if all the words in a webpage were to increase by one level of my measure, then one word in about six ads on the webpage would increase by one level. Changing one word in an ad can profoundly change the meaning of the ad. Consider changing the ad "Get your car price now!" (all top level 0 words) to the ad "Get your studebaker price now!" The second ad would be much more niche.

Because I repeatedly observe the same ad on the same webpage at different times, my results may be biased toward more frequent, less niche ads. Therefore, I also ran the regression (R1a) on unique observed ad-webpage combinations and removed any repeated observations of the same ad-webpage combination. I found that increasing the number of words in the webpage title decreases my measure y of ad niche-ness by 0.007. This result is significant at the 5% level. This would mean that if all the words in my webpage were to increase by one level of my measure, then one word in about ten ads on the webpage would decrease by one level. This significant difference in direction indicates that the relationship between webpage narrowness and ad niche-ness is not linear.

A significant amount of variation in my observations may depend on the daypart, as has been shown in other studies. For instance, Lowy (2003) showed significant differences in Internet audiences in gender, age, income level, size of audience, work / home use, and type of Internet use between five different dayparts: the early morning (Monday-Friday, 6am - 8am), the daytime (Monday-Friday, 8am - 5pm), the evening (Monday-Friday, 5pm - 11pm), the late night (Monday-Friday, 11pm - 6am), and the weekend (Saturday-Sunday). Therefore, I needed to check whether variations in my observations depended on the daypart as well. To test for this, I ran the regression

$$y = \beta_0 + \beta_1 x + \sum_{i=1}^{4} \beta_{\text{daypart}_i} t_{\text{daypart}_i} + \epsilon \qquad \text{(R2)}$$

where t_{daypart_1} is one if the observation was during the early morning, t_{daypart_2} is one if the observation was during the evening, t_{daypart_3} is one if the observation was during the

late night, and t_{daypart_4} is one if the observation was during the weekend. If all $t_{\text{daypart}_1} = t_{\text{daypart}_2} = t_{\text{daypart}_3} = t_{\text{daypart}_4} = 0$, then the observation was during the daytime. Therefore, a dummy variable for daytime was left out of the regression to avoid multicollinearity. In addition $\beta_{\text{daypart}_1}, \ldots, \beta_{\text{daypart}_4}$ are now additional parameters to be estimated, and the error term ϵ has now a conditional mean of zero given x, β_{daypart_1}, β_{daypart_2}, β_{daypart_3}, and β_{daypart_4}.

Running this regression (R2) on all observations, I find that observing my webpage during the early morning (compared to during the daytime) increases my measure y of ad niche-ness by 0.004. For all other times, ad niche-ness decreased: during the evening ad niche-ness decreased by 0.012, during the late night ad niche-ness decreased by 0.004, and during the weekend ad niche-ness decreased by 0.011. These results show significant differences in the ad niche-ness depending on daypart, with the exceptions of early morning not being significantly different than the daytime and evening not being significantly different than the weekend.

I also find that adding these daypart dummy variables does not change my estimates for the coefficient β_1, the relationship between ad niche-ness and webpage narrowness. I designed my experiment so that my data collecting program randomly observed my webpages across time. Therefore, daypart and webpage content are independent of each other.

I cannot run this regression on unique observed ad-webpage combinations because I observe the same ad-webpage combination at different times. Therefore I ran the regression

$$y = \beta_0 + \beta_1 x + \sum_{i=1}^{4} \beta_{\text{daypart}_i} \bar{t}_{\text{daypart}_i} + \epsilon \qquad \text{(R2a)}$$

where $\bar{t}_{\text{daypart}_i}$ is the average t_{daypart_i} for that particular ad-webpage combination. This should be seen as how frequently I see an ad-webpage combination at different times.

Running this regression (R2a) on unique observed ad-webpage combinations, I find that observing my webpage during the early morning (compared to the daytime) increases

my measure y of ad niche-ness by 0.014; during the evening ad niche-ness increases by 0.013; during the late night ad niche-ness increases by 0.001; and during the weekend ad niche-ness decreases by 0.009. These results show marginal differences in the ad niche-ness depending on daypart; namely, the evening and the weekend are significantly different from daytime at the 5% level.

Because my theory and my mixed linear results suggest that the relationship between webpage narrowness and ad niche-ness may not be monotonic, I also ran the following quadratic regressions

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon \qquad \text{(R3 and R3a)}$$

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \sum_{i=1}^{4} \beta_{\text{daypart}_i} t_{\text{daypart}_i} + \epsilon \qquad \text{(R4)}$$

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \sum_{i=1}^{4} \beta_{\text{daypart}_i} \bar{t}_{\text{daypart}_i} + \epsilon \qquad \text{(R4a)}$$

Running regressions (R3) and (R4) on all observations, I find a negative coefficient of -0.119 on the linear term and a positive coefficient of 0.103 on the quadratic term. This creates a U-shaped relationship between webpage narrowness and ad niche-ness: When webpage narrowness is below .578, increasing webpage narrowness decreases ad niche-ness, and when webpage narrowness is above .578, increasing webpage narrowness increases ad niche-ness. In other words: The least niche ads tend to appear more on the moderately-narrow webpages. These results were all significant at the .1% level for both regressions. In addition, this non-monotonicity did not change my estimates of the coefficients on the effects of dayparts in (R4) from the results I found in (R2).

Running regressions (R3a) and (R4a) on all unique observed ad-webpage combinations, I find a similar relationship; I find a negative coefficient of about -0.042 on the linear term and a positive coefficient of 0.027 on the quadratic term. This creates a U-shaped relationship between webpage narrowness and ad niche-ness: When webpage narrow-

ness is below .778, increasing webpage narrowness decreases ad niche-ness, and when webpage narrowness is above .778, increasing webpage narrowness increases ad niche-ness. In other words: The least niche ads tend to appear more on the moderately-narrow webpages. These results were also all significant at the .1% level for both regressions. In addition, this non-monotonicity did not change my estimates of the coefficients on the effects of dayparts in (R4a) from the results I found in (R2a).

5.6 Regression Estimates and Estimates

In this section, I examine my results from each regression run in Table 21 (b) and described in section 5.5. The purpose of this section is to ensure my regression does not estimate values outside the possible bounds of my measure of ad niche-ness. Because my measure is the average of the levels of words in an ad and because levels are between 0 and 2, I find that my measure of ad niche-ness is between 0 and 2. Therefore, in this section, I check whether my regressions give me estimates between 0 and 2.

Table 22 shows sample statistics for the regression estimates and residuals from the regressions in Table 21 (b) and described in section 5.5.

[Table 22 about here.]

In Table 22, the estimate of a regression is $\hat{y}$ where

$$\hat{y} \equiv max\{\min\{\hat{y}^*, 2\}, 0\} \tag{9}$$

$$\hat{y}^* \equiv \begin{cases} \hat{\beta}_0 + \hat{\beta}_1 x & \text{for (R1 and R1a)} \\ \hat{\beta}_0 + \hat{\beta}_1 x + \sum_{i=1}^{4} \hat{\beta}_{\text{daypart}_i} t_{\text{daypart}_i} & \text{for (R2)} \\ \hat{\beta}_0 + \hat{\beta}_1 x + \sum_{i=1}^{4} \hat{\beta}_{\text{daypart}_i} \bar{t}_{\text{daypart}_i} & \text{for (R2a)} \\ \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2 & \text{for (R3 and R3a)} \\ \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2 + \sum_{i=1}^{4} \hat{\beta}_{\text{daypart}_i} t_{\text{daypart}_i} & \text{for (R4)} \\ \hat{\beta}_0 + \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2 + \sum_{i=1}^{4} \hat{\beta}_{\text{daypart}_i} \bar{t}_{\text{daypart}_i} & \text{for (R4a)} \end{cases} \tag{10}$$

Here each $\hat{\beta}$ is the regression estimate of the true parameter β. And the residual is defined as the difference $y - \hat{y}$.

For regressions (R1) - (R4) or equivalently for regressions using all observations, I find my estimate have the same mean of 0.126 as my measure of ad niche-ness and my residuals have a mean of zero, which is what you would expect from an Tobit regression. The standard deviation of my measure y of ad niche-ness is 0.136, which is about the same as the standard deviation of the residuals $y - \hat{y}$. This comes from the fact that although my estimates are significant, the language in webpages and ads still has a lot of variation that my regressions do not explain. Each of the estimates $\hat{y}$ have a small standard deviation, and these estimates are all well within the bounds 0 and 2. They are all bounded below from about .11 and from above by about .3; none of the estimates are out of bounds of the possible measures of ad niche-ness.

I find similar results for regressions (R1a) - (R4a) or equivalently for regressions using unique observations of ad-webpage combinations. I find my estimate has the same mean of 0.124 as my measure of ad niche-ness and my residuals have a mean of about zero, which is what is expected from a Tobit regression. The standard deviation of my measure y of ad niche-ness is 0.146, which is about the same as the standard deviation of the residuals $y - \hat{y}$. Each of the estimates $\hat{y}$ have a small standard deviation, and these estimates are

also all well within the bounds 0 and 2; none of these estimates are out of bounds of the possible measures of ad niche-ness.

6 Conclusion

Using a Gibbs sampling algorithm on the HLDA model, I found strong evidence for a non-monotonic relationship: The least niche ads tend to appear more on the moderately-narrow webpages. Niche firms tend to value advertising on fine segments of consumers and mass advertising more than general products. Perhaps this comes from the set up of the auction where adjusting bids is based on the click-through rate or because only the more general firms can do market research on broad segments of consumers. In future research, I plan on addressing this issue by analyzing the affect of webpage narrowness on ad revenue.

prices of ads should answer this question.

Future research should incorporate the regression into the topic model. If we believe that webpage content could affect the creation of the ad's content, then it should be built into the topic model. The only reason that I did not do so here is so that this paper can serve as an example of the usefulness of topic model in economics that others can build upon to further the understanding of this dynamic field.

References

Agarwal, Nikhil, Susan Athey, and David Yang, "Skewed Bidding in Pay-per-Action Auctions for Online Advertising," *American Economic Review*, 2009, *99* (2), 441–47.

Agirgas, Cagdas, "What Drives Media Bias? A Panel Study of Newspaper Archives: 1990-2009," *Job Market Paper*, 2011.

Baeza-Yates, R. and B. Ribeiro-Neto, *Modern Information Retrieval*, New York: ACM Press, 1999.

Bergemann, Dirk and Alessandro Bonatti, "Targeting in advertising markets: implications for offline versus online media," *RAND J of Economics*, 2011, *42* (3), 417–443.

Blei, David M. and John D. Lafferty, "Dynamic Topic Models," in "ICML" 2006.

_ **and** _ , "Topic Models," in A. Srivastava and M. Sahami, eds., *Text Mining: Classification, Clustering, and Applications*, Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, 2009.

_ **, Andrew Ng, and Michael Jordan**, "Latent Dirichlet allocation," *JMLR*, 2003, *3*, 993–1022.

_ **, Thomas Griffiths, Michael Jordan, and Joshua Tenenbaum**, "Hierarchical topic models and the nested Chinese restaurant process," in "NIPS" 2003.

Chandra, Ambarish, "Targeted Advertising: The Role of Subscriber Characteristics in Media Markets," *The Journal of Industrial Economics*, 2009, *57* (1), 58–84.

Dickey, James M., "Multiple Hypergeometric Functions: Probabilistic Interpretations and Statistical Uses," *Journal of the American Statistical Association*, 1983, *78* (383), 628–637.

Gentzkow, Matthew and Jesse M. Shapiro, "Media Bias and Reputation," *Journal of Political Economy*, 2006, *114* (2), 280–316. Date revised - 2006-08-01; Language of summary - English; Pages - 280-316; ProQuest ID - 56516165; Corporate institution author - Gentzkow, Matthew; Shapiro, Jesse M; DOI - econlit-0859377; 0859377; 0022-3808.

Griffiths, Thomas L. and Mark Steyvers, "Finding Scientific Topics," *PNAS*, 2004, *101* (suppl. 1), 5228–5235.

Hofmann, Thomas, "Probilistic latent semantic analysis," in "UAI" 1999.

Iyer, Ganesh, David Soberman, and J. Miguel Villas-Boas, "The Targeting of Advertising," *Marketing Science*, 2005, *24* (3), 461 – 476.

Johnson, Justin P., "Targeted Advertising and Advertising Avoidance," 2013. forthcoming RAND J of Economics.

Larcinese, Valentino, Riccardo Puglisi, and Jr Snyder James M., "Partisan Bias in Economic News: Evidence on the Agenda-Setting Behavior of U.S. Newspapers," *Journal of Public Economics*, 2011, *95* (9-10), 1178–1189. Date revised - 2011-09-01; Language of summary - English; Pages - 1178-1189; ProQuest ID - 896012840; Corporate institution author - Larcinese, Valentino; Puglisi, Riccardo; Snyder, James M, Jr; DOI - econlit-1255076; 1255076; 10.1016/j.jpubeco.2011.04.006; 0047-2727.

Lewis, Randall A. and David H. Reiley, "Does Retail Advertising Work? Measuring the Effects of Advertising on Sales via a Controlled Experiment on Yahoo!," *Working Paper*, 2011.

_ **and** _ , "Advertising Effectively Influences Older Users: How Field Experiments Can Improve Measurement and Targeting," *Working Paper*, 2012.

Lowy, Lisa Sharkis, "The Existence and Characteristics of Dayparts on the Internet," *The OPA White Papers*, 2003, *1* (3).

Minka, Thomas and John Lafferty, "Expectation-Propagation for the Generative Aspect Model," *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, 2002, pp. 352–359.

Rosen-Zvi, Michal, Tom Griffiths, Mark Steyvers, and Padhraic Smyth, "The Author-Topic Model for Authors and Documents," in "UAI" 2004.

Steyvers, Mark and Tom Griffiths, "Probabilistic Topic Models," in T. Landauer, D. Mcnamara, S. Dennis, and W. Kintsch, eds., *Latent Semantic Analysis: A Road to Meaning.*, Laurence Erlbaum, 2006.

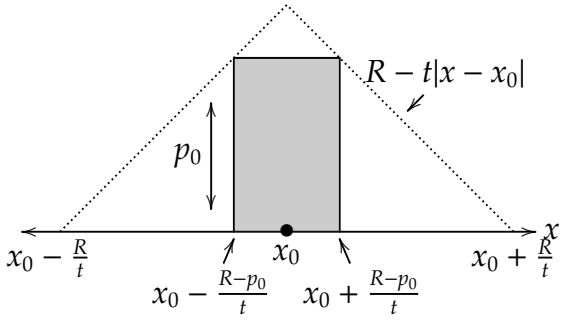
Teh, Yee-Whye, David Newman, and Max Welling, "A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation," in "NIPS" 2006.

Teh, Yee Whye, Michael I. Jordan, Matthew J. Beal, and David M. Blei, "Hierarchical Dirichlet Processes," *JASA*, 2006, *101*.

Wei, Xing and Bruce Croft, "LDA-based document models for ad-hoc retrieval," in "SIGIR" 2006.

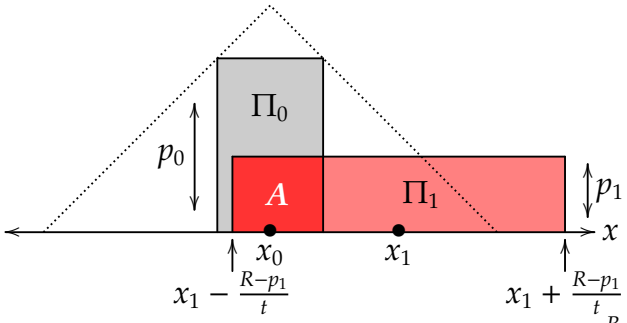
Figures and Tables

Figure 1: Standard Hotelling Model



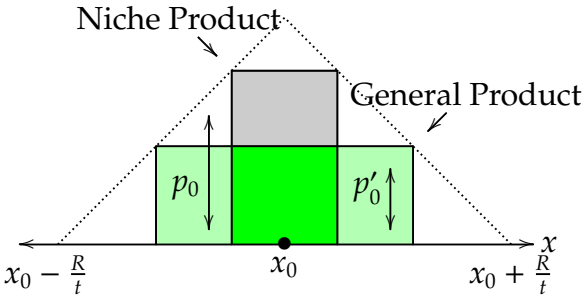
Note: For now price is exogenous (I relax this in Appendix A). The utility consumer x gets from buying from firm 0 is $u_0(x) \equiv R - t|x_0 - x| - p_0$. *Consumers in* $[x_0 - \frac{R-p_0}{t}, x_0 + \frac{R-p_0}{t}]$ *buy from firm 0. Firm 0's profit is* $p_0 * 2\frac{R-p_0}{t}$.

Figure 2: Hotelling Auction Duopoly: Example



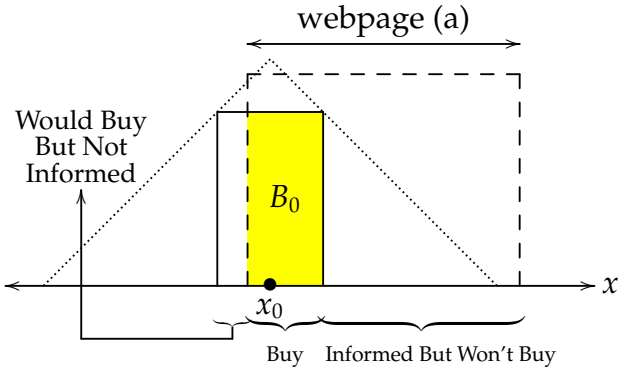
Note: A = *advertising cost to firm 0. Profit of firm 0 is* $p_0 * 2\frac{R-p_0}{t} - A = \Pi_0$.

Figure 3: Product Niche-ness vs. Price



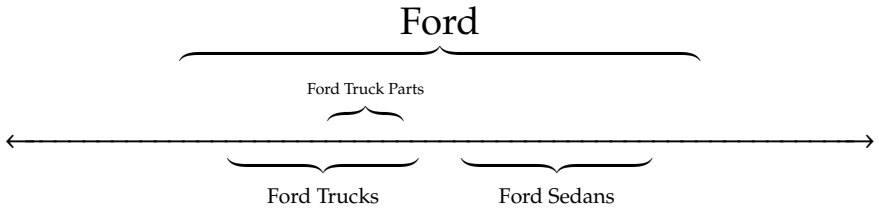
Caveat: In a full model, niche-ness is a function of consumer preferences, and price is a result of niche-ness. Here, price is a proxy of niche-ness.

Figure 4: Webpage Monopoly



Note: Firm 0's advertising bidding function b_0 is restricted to be constant for each webpage interval. This is the same as one bid: $B_0 = \int b_0(x)1\{x \in$ webpage interval$\}dx$.
Further note: The profit for firm 0 is B_0, because it is a monopolist.

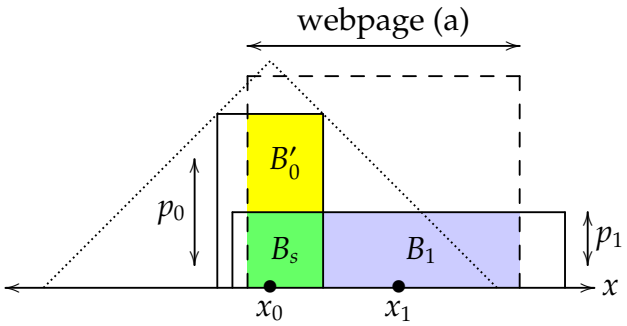
Figure 5: Webpage Title Segmentation



The number line $\mathbb{R}$ *is now split into webpage intervals. A 'Ford' webpage delivers a larger segment of consumers than a 'Ford Trucks' webpage. A 'Ford Trucks' webpage is narrower than a 'Ford' webpage.*
Note: Webpage narrow-ness is not related to product niche-ness. They can be different segments along the same number line.

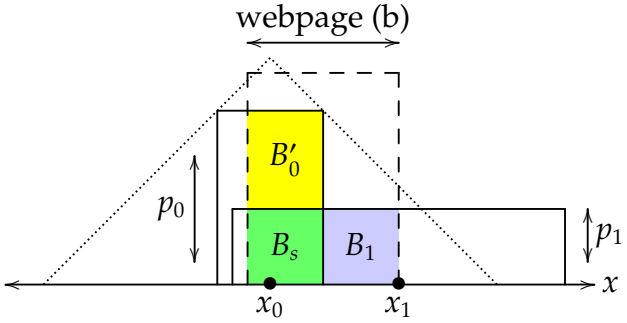
Figure 6: Hotelling Webpage Duopoly: Example

(a) General Product wins General Webpage



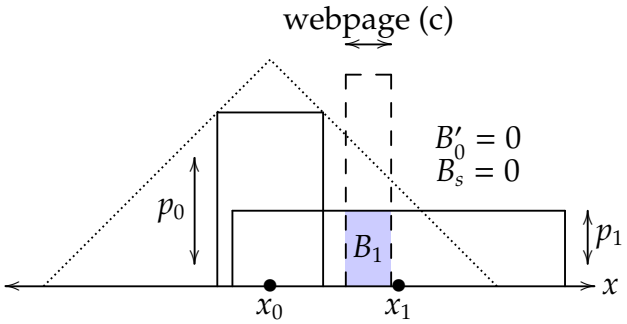
Note: Firm 0 is bidding $B_0' + B_s = B_0$. Firm 1 is bidding $B_1 + B_s$. $B_1 > B_0'$, so firm 1 wins the auction.

(b) Narrower Webpage



Note: $B_0' > B_1$, so firm 0 wins the auction.

(c) General Product wins Narrowest Webpage



$B_1 > B_0'$, so firm 1 wins the auction.

Figure 7: Example Webpage.

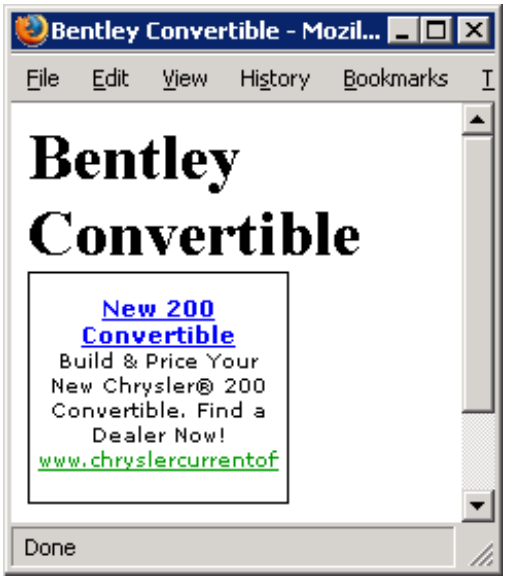


Table 8: Sample Data

	My Program.		From Flippa	Determined by Advertisers		
	Time and Date	**Webpage Content**	**Price**	**Ad URL**	**Ad Content**	
	...	...	...	...	...	
→	7/9/2013 3:07 PM	configuration diesels reference guide and turbocharger ...	$49	trucksmartsales.com	TruckSmart Service Center Light and Medium Duty Truck Service Diagnostics, Repairs, Inspections.	
	7/9/2013 3:07 PM	contact we pc market privacy custom this magazine ...	$499	davishyundai.com	Used Car Sale Trenton NJ Over 700 Vehicles In Our Inventory Come Find Yours Today.	← Same Ad
	7/9/2013 3:07 PM	citroen home mitsubishi ford daihatsu calculator ...	$130	davishyundai.com	Used Car Sale Trenton NJ Over 700 Vehicles In Our Inventory Come Find Yours Today.	← Same Ad
	...	...	...	...	...	
→	7/10/2013 11:09 AM	configuration diesels reference guide and turbocharger ...	$49	wholesaletransrepair.com	Transmissions Fixed Cheap 732-738-3834, Free Diagnostic. Financing Available, Free Loner Car.	
	...	...	...	...	...	

Same Webpage Different Observations

Note: I observe different ads on the same webpage at different times.
Further note: I also observe the same ad on different webpages.

Table 9: Top Words from 5 Topics Estimated by a Latent Dirichlet Allocation Algorithm

Topic 1: Computing & Servers	Topic 2: Tires & Cars	Topic 3: Shipping, Valuation, Education, & Cameras	Topic 4: Social Media & Hotrods	Topic 5: German
1. cloud	**1. manuals**	**1. shipping**	1. they	**1. der**
2. hosting	**2. tire**	**2. erase**	2. speedyrock	2. ford
3. trial	**3. tires**	**3. vin**	**3. wpturbo**	3. charting
4. netsuite	**4. barbie**	**4. 60%**	4. you	**4. und**
5. 15t	**5. michelin**	**5. adorama**	5. stroker	**5. emr**
6. pavilion	**6. 4-7**	**6. $100,000**	**6. wordpress**	**6. auf**
7. ftp	**7. wheels**	**7. valuation**	**7. facebook**	**7. sie**
8. dv6t	**8. honda**	**8. cordon**	8. $190	**8. oder**
9. dns	**9. rims**	**9. bleu**	**9. blogging**	9. degree
10. hp	**10. $69**	**10. rebel**	10. donate	**10. edd**
11. 17t	**11. mpg**	11. tablets	11. mastermind	**11. diese**
12. free	12. chip	**12. 3)$1**	**12. wiseco**	12. unlock
13. server	**13. mazda**	**13. 2)free**	13. april	**13. zu**
14. quad	**14. bmw**	14. car's	**14. plugin**	**14. umuc**
15. envy	15. selector	**15. 40%**	**15. bluehost**	15. torrent
16. crm	**16. horsepower**	**16. buys**	**16. empower**	16. dealer
17. vps	**17. toyota**	17. parts	**17. esb**	**17. zum**
18. backup	**18. subaru**	**18. educator**	**18. paving**	**18. eine**
19. marketing	**19. cadillac**	**19. dslr**	**19. piston**	**19. von**
20. tego	**20. bfgoodrich**	**20. campus-enroll**	**20. hotrods**	**20. sind**
21. management	**21. manual**	21. dns	**21. cobra**	**21. werden**
22. software	**22. nissan**	**22. reliability**	**22. gasket**	22. taurus
23. sampling	**23. freightliner**	**23. 5s**	23. 2,000	23. extendd
24. odbc	**24. dealer**	**24. cameras**	24. salvation	**24. den**
25. access	**25. suzuki**	25. 35%	25. table	**25. nicht**
26. i5-3230m	**26. volvo**	**26. skins**	26. nobody	**26. inhalte**
27. ghz	**27. delmarva**	27. jewelry	27. still	27. f-150
28. manage	28. prices	**28. bags**	28. bore	**28. seiten**
29. franchise	29. dart	29. cancer	**29. hosting**	**29. anfang**
30. download	**30. rover**	30. headphones	**30. kibblewhite**	**30. weiter**

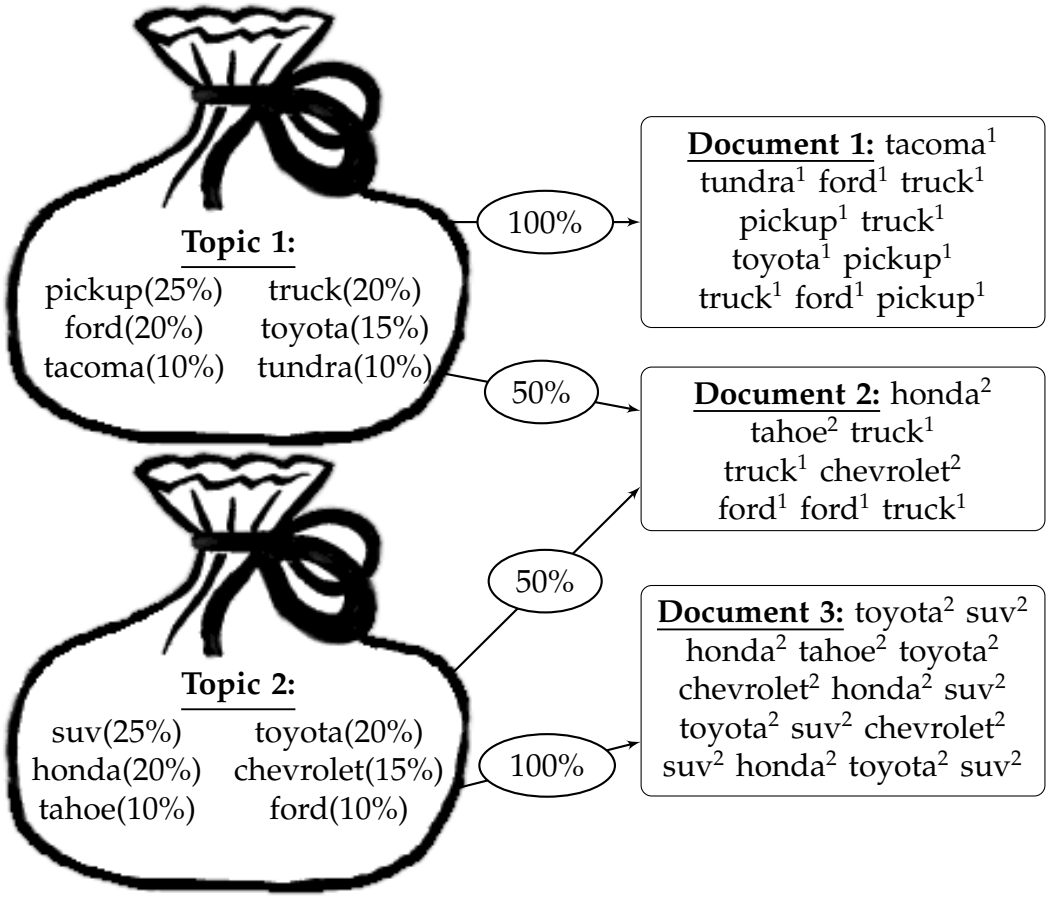
Topic names are created by the econometrician from looking at the top words in that topic. Top words are not the most probable words in a toipic, because the same word can appear in multiple topics. Top words are those with the highest term-score (from Blei and Lafferty, 2009, see (3)*).*

Table 10: Sample Ads with Estimated Topics from a Latent Dirichlet Allocation (LDA) Algorithm

PURE TOPIC 1 (Computing & Servers) AD: **Dashboard Analysis Powerful Business Software for Dashboarding and Scorecarding.**
MOSTLY TOPIC 2 (Tires & Cars) AD: **Pre-Owned VW Beetle Search Pre-Owned Inventory. See Special Offers on a VW Beetle.**
MIXED TOPIC 2 AND TOPIC 3 WEBSITE: cash cars we cars cash cash buy for hour cars cars
MIXED AD: **Mechanic Ripping You Off? Find Out Now From Our Experts. Ready To Chat. 100% Guaranteed**

In LDA, each word is a latent draw from a single topic, and different words in the same ad may be drawn from different topics.

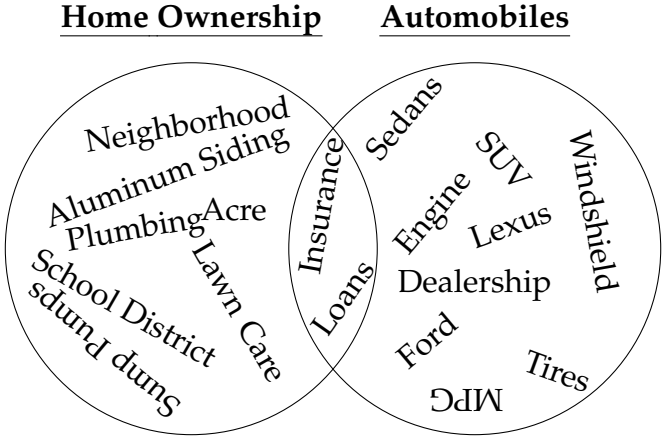
Figure 11: Latent Dirichlet Allocation Example



The proability of choosing a specific bag of words depends on the document (unique webpage or ad). Words are then selected for each document by first drawing a random bag of words, and then drawing a word from the bag of words.

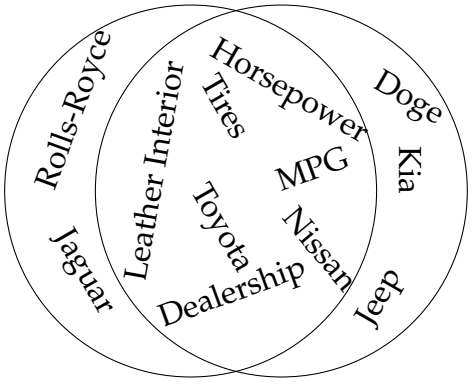
Figure 12: Identification of $\bar{\alpha}$

(a) Small Intersection Between Topics



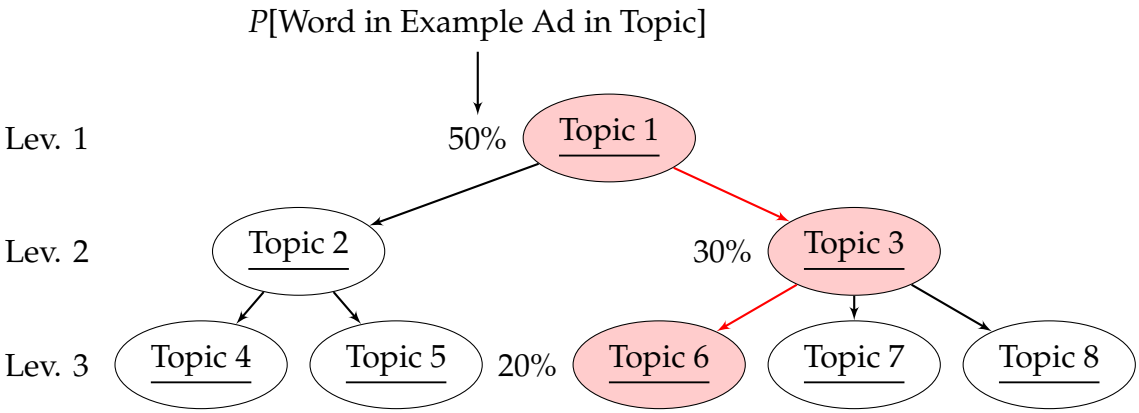
A small intersection between topics identifies a small $\bar{\alpha}$.

(b) Large Intersection Between Topics



A large intersection between topics identifies a large $\bar{\alpha}$.

Figure 13: Hierarchical Topics



The topic path chosen (the red filled in topics) is different from the set of probabilities of words chosen in each ad. The topic path only restricts the possible topics that a word can be drawn from. LDA is run on the smaller set of possible topics for each ad, after its topic path is chosen.

Table 14: Estimated Topic Allocation by γ (iteration 10,000).

The γ controls the probability of forming a new topic.

	γ = .2				γ = .25				γ = .3			
	Topic	**Parent**	**Docs**	**Words**	**Topic**	**Parent**	**Docs**	**Words**	**Topic**	**Parent**	**Docs**	**Words**
Level 0:	1	–	16,179	238,794	1	–	16,179	234,725	1	–	16,179	239,223
Level 1:	2	1	10,228	13,990	2	1	13,686	21,472	2	1	8,553	11,796
	3	1	1,633	2,943	3	1	727	8,424	3	1	2,745	8,402
	4	1	958	7,359	4	1	586	2,472	4	1	1,271	6,905
	5	1	924	3,048	5	1	407	2,862	5	1	1,270	2,465
	6	1	906	3,762	6	1	350	737	6	1	1,218	3,322
	7	1	849	2,101	7	1	247	2,213	7	1	706	1,210
	8	1	510	2,239	8	1	89	2,557	8	1	212	1,057
	9	1	133	2,552	9	1	65	127	9	1	155	2,594
	10	1	23	213	10	1	21	200	10	1	32	241
	11	1	8	23	11	1	1	1	11	1	8	13
	12	1	3	3					12	1	4	5
	13	1	3	4					13	1	4	3
	14	1	1	2					14	1	1	3
		Total:	16,179	38,239		Total:	16,179	41,065		Total:	16,179	38,016
Level 2:	15	2	10,172	1,698	12	2	13,116	3,063	15	2	8,440	1,377
	16	2	23	2	13	2	551	128	16	2	40	0
	17	2	8	5	14	2	11	1	17	2	39	2
	18	2	7	0	15	2	8	3	18	2	33	4
	19	2	6	1	16	3	350	237	19	2	1	0
	20	2	3	2	17	3	257	183	20	3	2,641	499
	21	2	3	1	18	3	81	43	21	3	68	11
	22	2	2	0	19	3	31	12	22	3	17	3
	23	2	2	1	20	3	8	6	23	3	17	1
	24	2	1	1	21	4	461	200	24	3	1	4
	25	2	1	1	22	4	120	43	25	3	1	0
	26	3	1,605	331	23	4	3	1	26	4	1,261	311
	27	3	16	3	24	4	2	2	27	4	10	1
	28	3	10	2	25	5	347	170	28	5	578	91
	29	3	2	1	26	5	46	68	29	5	535	108
	30	4	941	289	27	5	6	4	30	5	126	29
	31	4	8	0	28	5	6	1	31	5	11	1
	32	4	8	4	29	5	2	0	32	5	6	1
	33	4	1	0	30	6	104	27	33	5	6	3
	34	5	866	232	31	6	95	29	34	5	5	0
	35	5	29	9	32	6	94	36	35	5	2	0
	36	5	21	9	33	6	57	17	36	5	1	0
	37	5	5	3	34	7	151	49	37	6	1,123	261
	38	5	3	1	35	7	65	32	38	6	93	17
	39	6	607	109	36	7	16	4	39	6	2	1
	40	6	299	60	37	7	7	3	40	7	323	67
	41	7	680	224	38	7	4	3	41	7	321	68
	42	7	95	24	39	7	2	1	42	7	59	11
	43	7	67	20	40	7	1	1	43	7	3	2
	44	7	6	1	41	7	1	0	44	8	105	14
	45	7	1	1	42	8	30	8	45	8	96	53
	46	8	326	76	43	8	21	10	46	8	10	0
	47	8	165	22	44	8	20	4	47	8	1	1
	48	8	16	3	45	8	18	8	48	9	91	9
	49	8	3	0	46	9	54	15	49	9	47	6
	50	9	68	19	47	9	8	1	50	9	9	9
	51	9	65	18	48	9	1	0	51	9	4	0
	52	10	15	2	49	9	1	0	52	9	3	2
	53	10	8	8	50	9	1	1	53	9	1	0
	54	11	4	1	51	10	16	16	54	10	32	10
	55	11	3	1	52	10	2	1	55	11	8	4
	56	11	1	1	53	10	2	1	56	12	4	0
	57	12	2	0	54	10	1	0	57	13	2	2
	58	12	1	0	55	11	1	0	58	13	1	0
	59	13	2	1					59	13	1	0
	60	13	1	0					60	14	1	0
	61	14	1	2								
		Total:	16,179	3,189		Total:	16,179	4,432		Total:	16,179	2,983

γ = .25 produces few low-count topics, which means it manages to fit more of the ads to the common topics. In addition, it produces a good balance between number of words in each level and a more manageable number of topics.

Table 15: Number of Topics in Each Level with Low Word Count by γ (iteration 10,000)

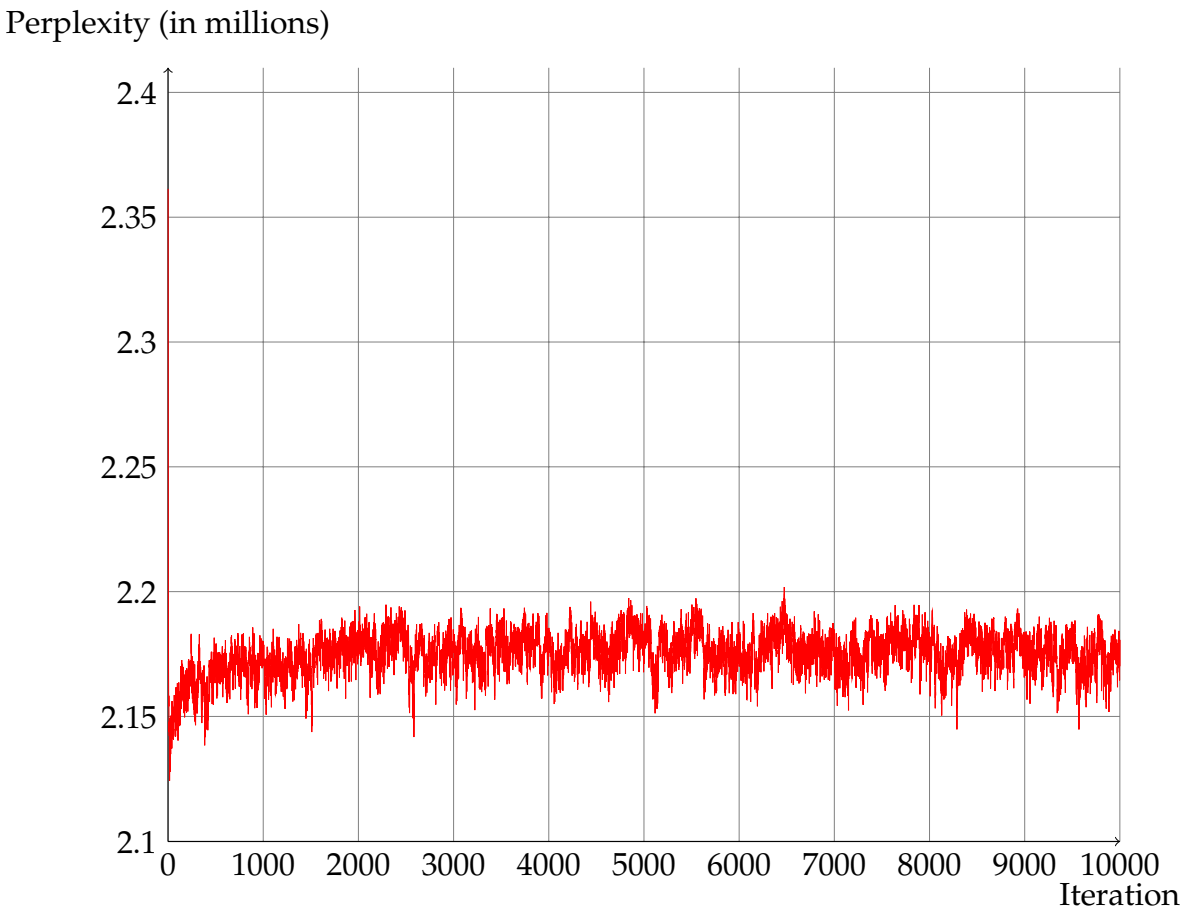
The γ controls the probability of forming a new topic.

Level	**Number of Topics with Word Count ≤**	$\gamma = 0.01$	$\gamma = 0.05$	$\gamma = 0.1$	$\gamma = 0.15$	$\gamma = 0.2$	$\gamma = 0.25$	$\gamma = 0.3$	$\gamma = 0.35$	$\gamma = 0.4$
0	∞	1	1	1	1	1	1	1	1	1
	50	0	0	0	0	0	0	0	0	0
	20	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
1	∞	14	11	14	14	13	10	13	15	12
	50	5	2	5	5	4	1	4	3	5
	20	5	1	4	3	3	1	4	2	5
	10	4	1	4	1	3	1	3	2	4
	5	2	0	1	1	3	1	3	2	4
	1	1	0	1	1	0	1	0	1	0
	0	0	0	1	0	0	0	0	0	0
2	∞	41	40	52	43	47	44	46	52	43
	50	35	32	44	37	39	37	37	45	34
	20	33	28	40	34	37	30	36	39	29
	10	28	24	38	31	34	26	32	33	25
	5	27	19	32	28	31	22	28	28	21
	1	19	13	22	22	21	15	19	22	12
	0	14	9	13	15	8	6	13	15	9

Word Count was calculated by taking the number of estimated words drawn from a topic (in each unique ad).

Note: $\gamma = .25$ produces few low count topics, which means it manages to fit more of the ads to the common topics. In addition, it produces a more manageable number of topics.

Figure 16: Convergence of Perplexity

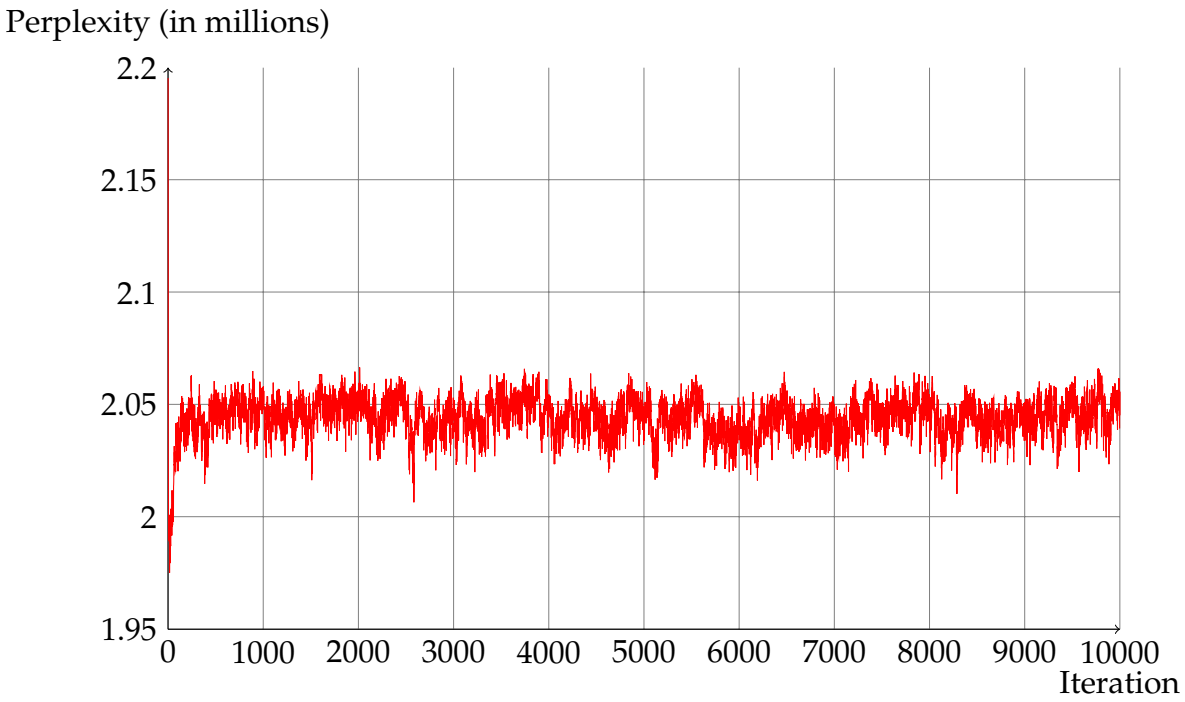


In language modeling, it is conventional to use perplexity instead of likelihood (see: Blei et al., 2003a). Perplexity is monotonically decreasing in the likelihood, and is equivalent to the inverse of the geometric mean of the per-word likelihood. A lower perplexity indicates a higher likelihood, or equivalently better performance.

Figure 17: Convergence of Estimates of $\vec{\eta}$

η_l is the symmetric Dirichlet parameter that determines the random draw of a probability $\beta_{k,v}$ of seeing a word v in a topic k from level l.

(a) $\vec{\eta}$ Perplexity



This is the perplexity conditional on the $\vec{\eta}$ parameters being true. In language modeling, it is conventional to use perplexity instead of likelihood (see: Blei et al., 2003a; Rosen-Zvi et al., 2004). Perplexity is monotonically decreasing in the likelihood, and is equivalent to the inverse of the geometric mean of the per-word likelihood. A lower perplexity indicates a higher likelihood, or equivalently better performance.

(b) $\vec{\eta}$ Values

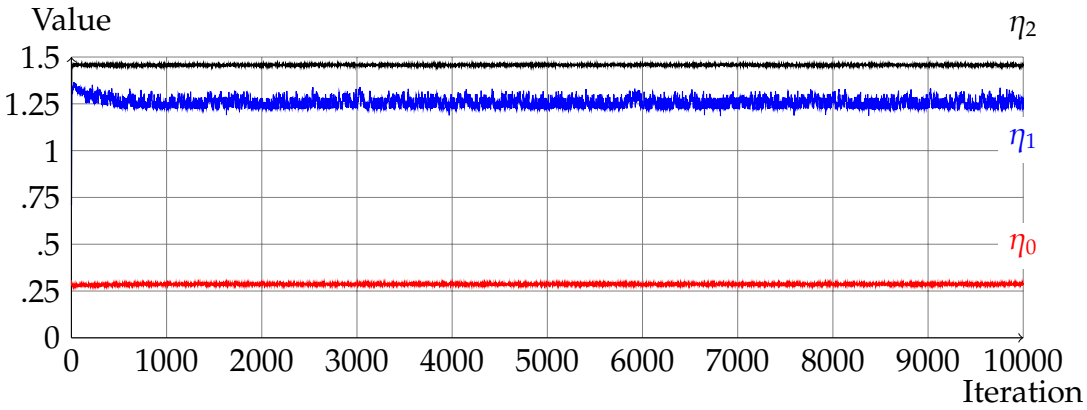


Table 18: Sample Frequency of Word Level Conditional on the Word Level of another Given Word in the Same Ad or Webpage

(a) Ads (All observations)

		Sample Frequency of Word (in ad)		
		Level 0	**Level 1**	**Level 2**
	Unconditional	88.60%	10.13%	1.27%
Conditional on a given word in same ad being	**Level 0**	89.82%	9.06%	1.15%
	Level 1	83.23%	15.07%	1.71%
	Level 2	83.28%	13.50%	3.26%

(b) Ads (Unique Ads)

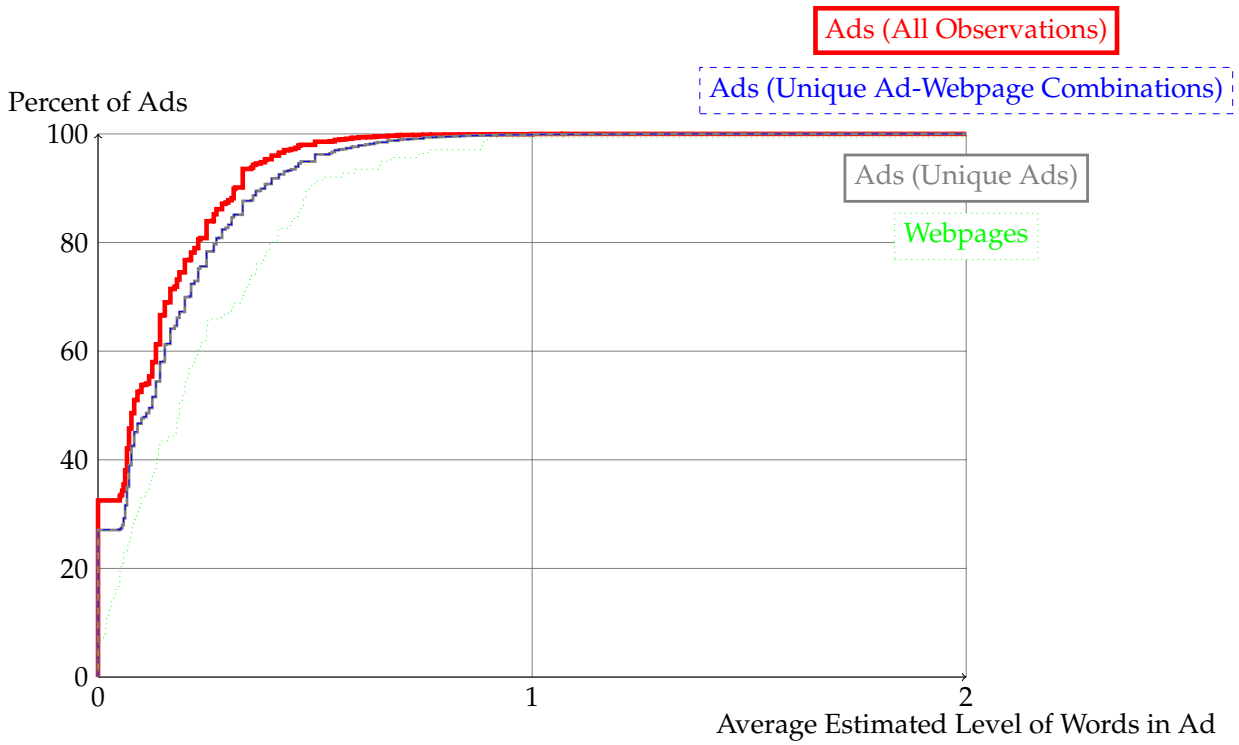
		Sample Frequency of Word (in ad)		
		Level 0	**Level 1**	**Level 2**
	Unconditional	86.05%	12.11%	1.85%
Conditional on a given word in same ad being	**Level 0**	87.89%	10.54%	1.59%
	Level 1	78.99%	18.41%	2.61%
	Level 2	78.39%	17.25%	4.38%

(c) Webpages

		Sample Frequency of Word (in webpage)		
		Level 0	**Level 1**	**Level 2**
	Unconditional	77.56%	20.19%	2.25%
Conditional on a given word in same webpage being	**Level 0**	82.85%	15.77%	1.38%
	Level 1	56.75%	41.31%	1.94%
	Level 2	69.27%	27.10%	3.62%

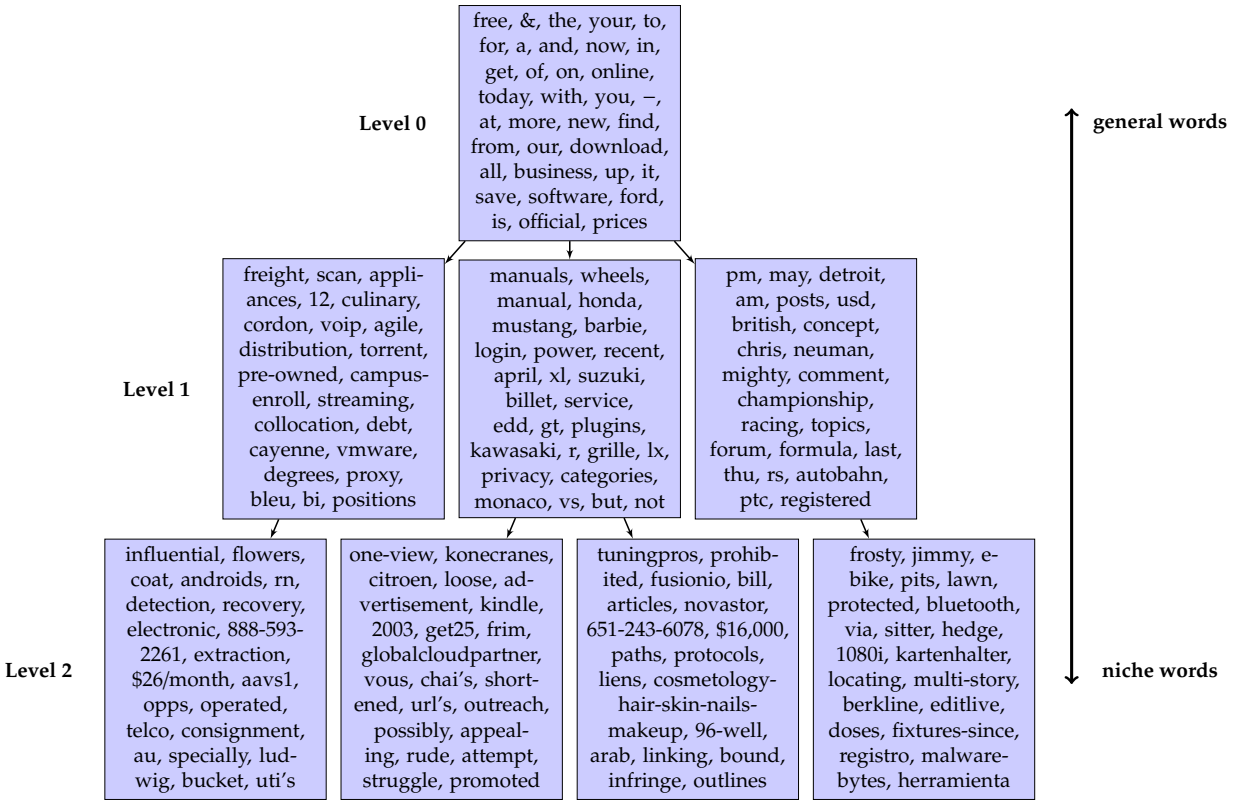
Note: These are data across unique webpage. Frequencies are the same (or very close) across all observations because there are roughly the same number of each webpage.

Figure 19: Cumulative Distribution Function of Average Estimated Level of Words in Ad



Levels of words were calculated by taking the mode of the estimated level from iterations 1,000, 2,000, 3,000, ..., and 10,000 of Blei et al. (2003b)'s Gibbs sampling algorithm for HLDA.

Figure 20: Top Words from Topics Identified by a Hierarchical Latent Dirichlet Allocation Algorithm



Each box represents a different topic identified by HLDA. Words in clusters or topics further down the tree are more specific; therefore, they ar more niche.

Note: Only the most common topics, or topics that occur frequently, are shown. There is only one level zero topic; it is shown. There are ten level one topics; the three topics shown are the only topics estimated to cover 2,862 or more words in my data (unique ads and webpages). There are forty-four level two topics; the four topics shown are the only topics that satisfy both: (1) have a parent topic of the three most common level one topics and (2) cover 170 or more words in my data.

Table 21: Results Using HLDA Measure of Niche-ness

Average level of words in ad estimated by the HLDA Algorithm.

(a) Sample Statistics on Ad Niche-ness & Webpage Narrowness

	Mean	StDev	Min	Max	N
Ad Niche-ness					
All Observations	.127	.136	0	1.333	893,614
Unique Ads	.158	.165	0	1.333	15,970
Unique Ad-Webpage Combinations	.123	.146	0	1.333	77,507
Webpage Narrowness					
All Observations	.247	.253	0	2	893,614
Unique Webpages	.247	.253	0	2	138
Unique Ad-Webpage Combinations	.251	.256	0	2	77,507

(b) Tobit Regressions of Webpage Narrowness on Ad Niche-ness

		All Observations				**Unique Ad-Webpage Combinations**			
		Linear Regressor		**Quadratic Regressors**		**Linear Regressor**		**Quadratic Regressors**	
		(R1)	(R2)	(R3)	(R4)	(R1a)	(R2a)	(R3a)	(R4a)
Webpage	Narrowness	0.012***	0.012***	-0.119***	-0.119***	-0.007*	-0.006*	-0.042***	-0.041***
		(0.001)	(0.001)	(0.002)	(0.002)	(0.003)	(0.003)	(0.006)	(0.006)
	Narrowness^2	—	—	0.103***	0.103***	—	—	0.027***	0.027***
				(0.001)	(0.001)			(0.004)	(0.004)
Time of Day	Early Morning	—	0.004	—	0.003	—	0.014	—	0.014
			(0.002)		(0.002)		(0.008)		(0.008)
	Evening	—	-0.012***	—	-0.012***	—	0.013*	—	0.013*
			(0.001)		(0.001)		(0.005)		(0.005)
	Late Night	—	-0.004***	—	-0.004**	—	0.001	—	0.001
			(0.001)		(0.001)		(0.005)		(0.005)
	Weekend	—	-0.011***	—	-0.011***	—	-0.009*	—	-0.008*
			(0.001)		(0.001)		(0.004)		(0.004)
	Constant	0.084***	0.094***	0.104***	0.114***	0.074***	0.080***	0.080***	0.085***
		(0.000)	(0.001)	(0.000)	(0.001)	(0.001)	(0.004)	(0.001)	(0.004)
	σ	0.186***	0.186***	0.185***	0.185***	0.207***	0.207***	0.207***	0.207***
		(0.000)	(0.000)	(0.000)	(0.000)	(0.001)	(0.001)	(0.001)	(0.001)
	p	0.0000	0.0000	0.0000	0.0000	0.0309	0.0000	0.0000	0.0000
	N	893,614	893,614	893,614	893,614	77,507	77,507	77,507	77,507

* = 5% *significance,* ** = 1% *significance, and* *** = .1% *significance.*

Note: Results were similar for OLS regressions.

Table 22: Residuial Sample Statistics for Regressions

	Mean	**St. Dev.**	**Min.**	**Max.**
All Observations (N = 893,614)				
"True Value"	0.127	0.136	0.000	1.333
Estimate (R1)	0.126	0.002	0.124	0.140
Estimate (R2)	0.126	0.003	0.122	0.150
Estimate (R3)	0.126	0.016	0.114	0.284
Estimate (R4)	0.126	0.016	0.112	0.296
Residual (R1)	0.001	0.136	−0.140	1.209
Residual (R2)	0.001	0.136	−0.150	1.210
Residual (R3)	0.001	0.135	−0.284	1.220
Residual (R4)	0.001	0.135	−0.296	1.216
Ad-Webpage Combinations (N = 77,507)				
'True Value"	0.123	0.146	0.000	1.333
Estimate (U1)	0.124	0.001	0.117	0.125
Estimate (U2)	0.124	0.004	0.115	0.138
Estimate (U3)	0.124	0.004	0.118	0.146
Estimate (U4)	0.124	0.005	0.117	0.159
Residual (U1)	−0.001	0.146	−0.125	1.211
Residual (U2)	−0.001	0.146	−0.138	1.211
Residual (U3)	−0.001	0.146	−0.146	1.215
Residual (U4)	−0.001	0.146	−0.159	1.213

For regressions see table 21 (b). "True Value" refers to the value of ad niche-ness estimated by the HLDA algorithm. Estimate (x) refers to the estimated value of ad niche-ness and Residual (x) refers to the residual using regression (x).

Appendix A Many Firms / Endogenous Prices

Now let's consider a case of where there are N firms bidding in an auction for an auction for a single webpage. Let it be common knowledge that the center of the webpage interval be located at $x_w = 0$ and the length of its interval be $\ell > 0$. Let each firm $j = 1, ..., N$ receives a random, privately-known, horizontal product characteristic or location of $x_j \sim U[-M, M]$, where $M > \ell$. Then the quantity q_j of consumers delivered by the webpage that would buy product $j = 1, 2, 3, ..., N$ at the price p_j would be given by equation (11). Giving a first-order pricing condition of equation (12).

$$q_j(p_j, \ell, |x_j|) = \begin{cases} 0 & \text{if } |x_j| \geq \ell + \frac{R-p_j}{t} \\ 2\ell & \text{if } |x_j| \leq \frac{R-p_j}{t} - \ell \\ 2\frac{R-p_j}{t} & \text{if } |x_j| \leq \ell - \frac{R-p_j}{t} \\ \frac{R-p_j}{t} + \ell - |x_j| & \text{otherwise} \end{cases} \tag{11}$$

$$p_j^*(\ell, |x_j|) = \begin{cases} \frac{R}{2} & \text{if } |x_j| \leq \ell - \frac{R}{2t} \\ R - t(\ell - |x_j|) & \text{if } \ell - \frac{R}{2t} \leq |x_j| \leq \ell - \frac{R}{3t} \\ \frac{R+t(\ell-|x_j|)}{2} & \text{if } \ell - \frac{R}{3t} \leq |x_j| \leq \ell = \frac{R}{t} \\ \text{anything} & \text{if } |x_j| \geq \ell = \frac{R}{t} \end{cases} \tag{12}$$

It is straight forward to show using equation (12) that the closest firm to $x_w = 0$ would win the auction. From this result and equation (12), the closest firm (and therefore the winner of the advertising auction) may be a lower-priced, more general product or a higher-priced, more niche product. In general, we can simply say that firm $j =$ is niche if $\frac{R-p_j}{t} > \frac{R-\underline{p}}{t}$ or equivalently if $p_j > \underline{p}$. Let $underbarp > R/2$.[48] It follows that the winning advertiser is niche when $\ell - \frac{R-p}{t} < |x_j| < \ell = \frac{R-2p}{t}$; therefore, the closest and furthest firms from the webpage sell general products, while the niche products are sold in a band

[48] If $underbarp < R/2$, then the winning advertiser is niche when $|x_j| < \ell - \frac{R-2p}{2t}$; therefore the closest firms sell the niche products. This is an uninteresting case.

around $x_w = 0$.

The probability that at least one firm is within $x \in [0, M]$ of $x_w = 0$ is $1 - (1 - \frac{x}{M})^N$. I consider the limit case where there are many firms and M is huge; I take M and N to infinity and hold $\lambda \equiv \frac{N}{M}$ constant. This probability that at least one firm is within $x \in [0, M]$ of $x_w = 0$ becomes $1 - e^{-\lambda x}$. Therefore, the probability density function of the closest firm to $x_w = 0$ (and therefore the winner of the advertising auction) is $\lambda e^{-\lambda x}$. The probability that the closest firm to $x_w = 0$ (and therefore the winner of the advertising auction) is a niche product is given by equation (13).

$$Pr(\text{Niche Firm Wins}) = \begin{cases} e^{-\lambda \ell}(e^{\lambda \frac{R-p}{t}} - e^{-\lambda \frac{R-2p}{t}}) & \text{if } \ell > \frac{R}{2t} \\ 1 - e^{-\lambda(\ell + \frac{R-2p}{t})} & \text{if } \ell < \frac{R}{2t} \end{cases} \tag{13}$$

Notice when $\ell > \frac{R}{2t}$ or equivalently when the webpage is generally-focused, as a webpage becomes more narrowly-focused (as ℓ shrinks), a niche product would have an increasing chance of winning the auction. Additionally when $\ell < \frac{R}{2t}$ or equivalently when the webpage is narrowly-focused, as a webpage becomes more narrowly-focused (as ℓ shrinks) a niche product would have a decreasing chance of winning the auction. This shows that the relationship between the niche-ness of the advertised product and the narrowness of the webpage is not necessarily monotonic, and therefore is an empirical question.

To empirically test which kind of firm win the auction for advertising on a webpage, I regress ad niche-ness (my independent variable) on webpage narrowness (my dependant variable). If I knew the webpage narrowness and the relative locations of the firms and webpages, then I could predict perfectly which kind of firm (niche or general) would win the auction to advertise on a webpage, because I would be able to solve the firm's problem for pricing, ignoring any multiple equilibria or the influence of outside markets on price. However, because I do not know locations and pricing, I have an error in the kind of firm that wins the auction for advertising, so I need to investigate which type of firm tends to win the bid for advertising on which type of webpage.