# Improving Classification Accuracy Using Automatically Extracted Training Data

Ariel Fuxman
Microsoft Research
Mountain View, CA, USA
arielf@microsoft.com

Anitha Kannan
Microsoft Research
Mountain View, CA, USA
ankannan@microsoft.com

Andrew B. Goldberg*
Univ. of Wisconsin-Madison
Madison, WI, USA
goldberg@cs.wisc.edu

Rakesh Agrawal
Microsoft Research
Mountain View, CA, USA
rakesha@microsoft.com

Panayiotis Tsaparas
Microsoft Research
Mountain View, CA, USA
panats@microsoft.com

John Shafer
Microsoft Research
Mountain View, CA, USA
jshafer@microsoft.com

## ABSTRACT

Classification is a core task in knowledge discovery and data mining, and there has been substantial research effort in developing sophisticated classification models. In a parallel thread, recent work from the NLP community suggests that for tasks such as natural language disambiguation even a simple algorithm can outperform a sophisticated one, if it is provided with large quantities of high quality training data. In those applications, training data occurs naturally in text corpora, and high quality training data sets running into billions of words have been reportedly used.

We explore how we can apply the lessons from the NLP community to KDD tasks. Specifically, we investigate how to identify data sources that can yield training data at low cost and study whether the quantity of the automatically extracted training data can compensate for its lower quality. We carry out this investigation for the specific task of inferring whether a search query has commercial intent. We mine toolbar and click logs to extract queries from sites that are predominantly commercial (e.g., Amazon) and non-commercial (e.g., Wikipedia). We compare the accuracy obtained using such training data against manually labeled training data. Our results show that we can have large accuracy gains using automatically extracted training data at much lower cost.

## Categories and Subject Descriptors

H.2.8 [**Database management**]: Database applications - data mining

## General Terms

Algorithms, Experimentation

*Work done when the author interned at Microsoft Research.


*KDD'09,* June 28–July 1, 2009, Paris, France.


## 1. INTRODUCTION

Classification lies at the core of many knowledge discovery and data mining (KDD) applications whose success depends critically on the quality of the classifier. There has been substantial research in developing sophisticated classification models and algorithms with the goal of improving classification accuracy, and currently there is a rich body of such classifiers.

On the other hand, there is work coming out of the NLP community that suggests that for tasks such as natural language disambiguation a simple algorithm can outperform a sophisticated algorithm if it is provided with more training data [4, 10, 13]. In the application settings considered in these papers, the training data occurs *naturally* in text corpora, and *high quality* training data sets running into billions of words have been used.

This paper explores how we can apply the lessons from the NLP community in KDD settings. In particular, we address the question of how to obtain massive labeled data sets cheaply. Contrary to the NLP setting, such almost-for-free data is not readily available, and it is often noisy. We investigate whether we can still get high accuracy when there is no one good source for training data, and the extracted data contains inconsistencies.

To keep the discussion concrete, we consider the specific task of inferring whether a query posed to a search engine has commercial intent (i.e., the user plans to buy a tangible product). Because of large variations in the search queries, success in this task requires large amounts of training data—queries labeled as commercial or non-commercial. However, manual labeling of queries is time-consuming and expensive. As the queries change with the passage of time, new labeling is constantly required. Consequently, there is never enough training data. Moreover, there is often inconsistency in the labels assigned even by experts [1].

In our study, we extract training data by mining toolbar and click logs. We collect queries posed to commerce-focused portals (Amazon, Craigslist) as the source of positive examples. We also mine negative examples from click logs by obtaining queries that frequently lead to non-commercial portals (Wikipedia). Several issues arise immediately:

- It is difficult to find portals that are purely commercial or non-commercial. For instance, there are pages about Barbie in Wikipedia, and Amazon sells books

and movies called "World War I". In general, how do we ensure the quality of the labels assigned to the extracted data?

- Should we use queries from Amazon, or Craigslist, or both? In general, given a large number of possible data sources, which are the best sources to use for the classification task at hand?

- Should we use all the examples we can mine from a data source, or are there some that are better than others?

We address these issues and show how to leverage good but imperfect sources to quickly and cheaply generate massive amounts of training data as frequently as needed. In the end, we show that this data yields high accuracy classification results. Our findings are applicable to any domain where it is possible to find data sources that are related to the target class. For example, consider the task of deciding if a query has "local" intent (i.e., it reflects the intent to purchase goods from a local store). We could then use a site such as YellowPages.com as a source of local queries.

As we have already argued, for most classification tasks obtaining large amounts of manually labeled training data is expensive and often simply infeasible. The problem of training data sparsity has been studied in the machine learning community, where two research directions have been pursued: i) designing complex feature representations to remedy feature sparseness, and ii) leveraging unlabeled data to compensate for the limited amounts of labeled data [17]. The latter approaches are often collectively referred to as semi-supervised learning (SSL).

We wish to draw a distinction between SSL and our approach of automatically extracting large amounts of labeled data. SSL starts with as much manually labeled data as it is feasible to obtain. SSL then tries to leverage unlabeled data, which is assumed to be available in abundance for free, to learn a better classification model. While SSL algorithms may predict labels for the unlabeled data either during or after the learning process, the quality of these predicted labels depends strongly on the amount of initially labeled data. If little to no reliable labeled data is initially available, an SSL approach may not succeed, even if it has access to large amounts of unlabeled data. In contrast, the goal of our approach is to select the right data sources from which labeled data can be extracted quickly and in a straightforward manner. In fact, the labels are known inherently from the sources which guarantees obtaining large amounts of labeled data with ease. Thus, our approach does not require any manually labeled data but instead aims at selecting the sources from which to obtain labeled data. In the empirical evaluation in Section 5 we compare our approach against an SSL approach, namely a self-training algorithm.

The rest of the paper is organized as follows. In Section 2, we describe our methodology for collecting automatically labeled data: selecting data sources, extracting data, and identifying those data points that are useful for training. In Section 3 we describe the problem of commercial intent identification for search engine queries. In Section 4, we show how we employed the methodology presented in Section 2 to automatically extract labeled data for the commercial intent identification task. Section 5 presents an empirical evaluation of the approach, including a comparison against classifiers trained on manually labeled data. Section 6 reviews related work, and Section 7 gives concluding remarks.

## 2. IDENTIFYING SOURCES OF TRAINING DATA

In this section, we discuss desirable properties of potential sources of training data, followed by a method for selecting which data from those sources should be extracted and automatically labeled.

We propose that the sources of training data should satisfy the following properties:

- *Popularity*: The sources should be popular because only then can they yield large amounts of data.

- *Orthogonality*: The sources should provide training data about different regions of the training space.

- *Separation*: The sources should provide either positive or negative examples of the target class, but not both.

We now describe how we can use these properties as a guide for selecting good sources of training data.

**Popularity.** The popularity property essentially states that among different possible sources we should select those that contain the largest amount of data. For example, if we are interested in extracting tagged images for image classification, we should extract images from sites like Flickr[1], one of the popular destinations on the web for posting and tagging images. For the commercial intent classification task, we may want to extract queries posed to popular commercial portals such as Amazon.

For many classification tasks, it is possible to resort to publicly available statistics in order to choose sources that satisfy the popularity requirement. Internet survey companies such as Hitwise[2] provide Web traffic reports, which can be used for selecting popular sources. In Section 4 we describe how we selected the popular sources for the commercial intent classification task.

**Orthogonality.** The orthogonality property essentially states that the sources we select should be diverse so that they provide different types of training data. For example, for the commercial intent classification task, we want to have examples related to different types of products that a user may query about. One data source may be good in providing training examples of queries for electronics products, while another one may be good for queries on used furniture. Again, we can use external sources for identifying orthogonal sources. Web directories, or Internet survey companies such as Hitwise categorize Web sites according to the services they provide. We can make use of this categorization in order to select sources that span different categories.

**Separation.** The separation property states that the training examples must unambiguously reflect the intended meaning of their source. For example, the query "World War I" can be commercial if it is posed to the Amazon portal, where it refers to a book. However, it is clearly not commercial in any other context. We now propose a technique to enforce

[1] `http://www.flickr.com`
[2] `http://www.hitwise.com`

the separation property, which relies on the fact that the feature space of two distinct classes is different.

Our focus is on classification problems where the target class is relatively rare. In other words, the positive class represents a concept for which we are given a definition (e.g., commercial intent), while the negative class includes everything else. Thus, our technique for enforcing separation is based on conservatively refining the set of positive examples, while using as many negative examples as possible.

The first step of the technique is to obtain candidate positive examples from the positive sources and negative examples from the negative sources. For instance, queries from Amazon and Wikipedia can serve as positive and negative examples, respectively, for commerce intent classification task. In order to make refinement easier, the second step is to separate the candidate positive examples into groups if possible (e.g., by clustering or using available metadata), and compute the frequency distribution of features for each group. The last step is to compare the distributions of each group against the distribution of the negative examples, and keep only those groups whose distribution is highly divergent with respect to the negative distribution. Groups of examples too similar to the negative class are discarded. Note that for some applications or some data sources, it may not be sensible to subdivide the candidate positive examples into groups. In this case, we can compare the distribution of the entire set of positive examples to the distribution of the negative examples.

The above refinement process involves comparing feature frequency distributions between sets of examples. The frequency distribution of features, $p(w|S)$, in a set $S$ (e.g., a group of candidate positive examples or all negative examples) is defined as the fraction of times that the feature appears in that set:

$$p(w|S) = \frac{\text{number of occurrences of } w \text{ in set } S}{\text{total count of features in set } S}.$$

We measure similarity between distributions using Jensen-Shannon (JS) divergence. This symmetrized and smoothed version of the Kullback-Leibler (KL) divergence [6] provides a good estimate of the true divergence, as it takes into account the non-overlapping features of the two distributions under consideration. The KL-divergence between two distributions $P$ and $Q$ is computed as:

$$KL(P||Q) = \Sigma_w P(w) \log \frac{P(w)}{Q(w)},$$

and the corresponding The Jensen-Shannon divergence is defined as:

$$JS(P,Q) = \frac{1}{2}(KL(P||M) + KL(Q||M)),$$

where $M$ is the "average distribution" computed as $M = \frac{1}{2}(P+Q)$. In Section 4, we provide an example to illustrate the refinement process based on JS divergence.

# 3. COMMERCIAL INTENT IDENTIFICATION TASK

For the investigation of the central thesis of the paper, we choose to study the task of identifying search queries with commercial intent. From a technical point of view, this task is interesting for the following reasons:

- It is inherently a difficult task. In fact, it is not obvious to even precisely specify which queries have commercial intent. We can at best give examples, and learn the rough definition inductively. Based on a preliminary analysis of example queries, we arrived at the informal definition of commercial intent listed below.

- Only a small fraction of queries (between 5 and 10%) are deemed to be commercial. Thus, sampling the log of search queries and employing human annotators to label commercial queries is particularly time-wasting and expensive as the annotators end up spending most of their time labeling non-commercial queries.

- Unlike some tasks where one might be able to identify an obvious, perfect source for obtaining labeled data (e.g., the text or Web corpora for NLP algorithms [4, 10, 13]), there is no such source for this task. It thus requires us to understand how to use sources that are an imperfect approximation of the true labeled data. At the same time, the Web is home to many commercial portals that can yield a large number of queries. However, care should be taken to ensure that we obtain high quality training data.

Additionally, this task has important practical value. A large fraction of online commercial transactions are initiated with a query to a generic search engine. Thus, in order to customize the user experience around shopping, it is fundamental for the search engines to detect the commercial intent of their users.

### Informal Definition of Commercial Intent

For the purposes of this paper, we say that a query has commercial intent if most of the users who type the query have the *intention to buy* a *tangible product*. Examples of tangible products include items such as books, furniture, clothing, jewelry, household goods, vehicles, etc. Services are not included.For example, "medical insurance" and "cleaning services almaden" are queries that are not considered commercial.

Intention to buy means that the user intends to perform a commercial transaction in which she will acquire the product. It includes cases in which a user researches the product before buying it. For example, "digital camera reviews" and "digital camera price comparison" reflect intention to buy. Intention to buy also means that money will be spent on the product and thus excludes products that can be obtained for free. For example, "free ringtones" does not have commercial intent because the user wants to get the product for free.

It is apparent that this definition is very broad and rather ambiguous, which adds to the complexity of the task.

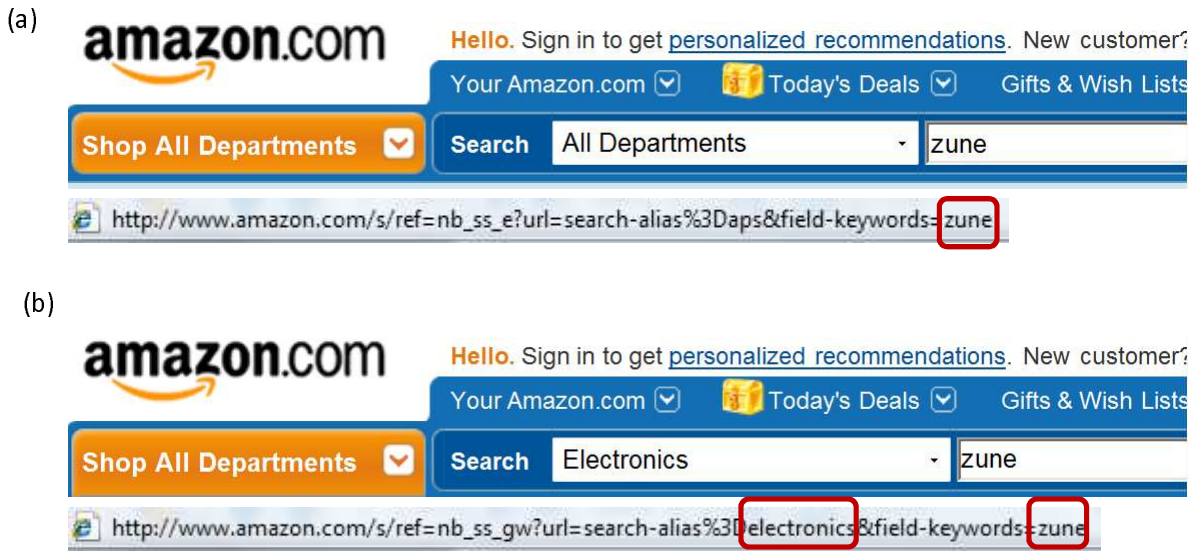


**Figure 1: URLs recorded in the toolbar logs. (a) Query "zune" posed on Amazon site results in a URL that has query "zune" as a keyword. (b) When the same query is posed after choosing a category, we extract both the category (in this case, "Electronics") and the query.**

## 4. LABEL GENERATION FOR COMMERCIAL INTENT

In this section, we show how we applied the label generation process to the commercial intent identification task. We show the sources that we chose; we justify why they satisfy the properties that we propose in Section 2; and we present the methods that we used to extract commercial queries from toolbar logs and non-commercial queries from click logs.

### Choosing Data Sources

We used the Hitwise Web traffic report as the basis for identifying data sources for the commercial intent identification task. Based on this report, we chose Amazon and Craigslist as sources of commercial queries, and Wikipedia as a source of non-commercial queries. We can use the Hitwise report to justify why these sources satisfy the desired properties. First, they are popular. According to Hitwise's September 2007 survey, Amazon has 29% of the entire Web traffic in the "Department Stores" category; Craigslist has 52% of the traffic in the "Classified Ads" category; and Wikipedia has 27% of the traffic in the "Reference" category. Second, Amazon and Craigslist have considerable degree of orthogonality because the kinds of products offered by "Classified Ads" and "Department Stores" are quite different. For example, Amazon provides a wide array of electronic products, while Craigslist does not have so many of them. Craigslist does have many offers about cars, however, which Amazon lacks. Finally, Craigslist and Amazon are likely to be separable from Wikipedia since the former focuses on commercial content, while the latter focuses on informational and mostly non-commercial content.

### Extracting Commercial Queries

We extracted the commercial queries from the toolbar log of a popular Web browser. We chose to use the toolbar log for two reasons. First, the queries that users pose at a portal are influenced by the type of content the site serves. Thus, if a user accesses a commercial portal and types a query in the search box, then she is likely to have commercial intent. We will shortly show how we used the toolbar logs to obtain the queries that are typed *directly* on the search box of commercial portals such as Amazon and Craigslist. Second, we can use the toolbar logs to obtain additional metadata associated with the queries. In particular, we extracted queries together with category assignments (i.e., the sales departments related to the queries). The categories were used to divide the queries into groups and apply the technique presented in Section 2. They were also used to discard groups that were not fully aligned with the definition of commercial intent (e.g, the category "Tickets" in Craigslist).

To illustrate our use of the toolbar logs, consider a user who types the query "zune" on the search box of Amazon (Figure 1a). Once the query is submitted, the following URL is generated:

`http://amazon.com/...field-keywords=zune`

It is easy to see that the query can be directly parsed out from the URL. Thus, given access to the URLs that record the activity of the users on the Amazon site, we can extract all the queries that are typed in the search box. All search engine companies provide browser toolbars and save toolbar logs that record such URLs.

It is also possible to use the toolbar logs to obtain the categories associated with the queries, which can be used to separate the queries into groups. Continuing with the "zune" example, suppose that the user selects the category "Electronics" before typing "zune" (Figure 1b). Then, the following URL is generated:

`http://amazon.com/...electronics&field-keywords=zune`

Again, both the category and the query can be extracted from the URL.

| Amazon | |
|---|---|
| Category | JS |
| Photo | 0.580 |
| Computers | 0.579 |
| Automotive | 0.531 |
| Tools | 0.524 |
| Electronics | 0.518 |
| . | . |
| . | . |
| . | . |
| VHS | 0.347 |
| Music | 0.340 |
| DVD | 0.314 |
| All Departments | 0.307 |
| Books | 0.288 |

| Craigslist | |
|---|---|
| Category | JS |
| Motorcycles/Scooters | 0.563 |
| Auto Parts | 0.558 |
| Photo and Video | 0.553 |
| Music Instruments | 0.551 |
| Tools | 0.549 |
| . | . |
| . | . |
| . | . |
| General | 0.491 |
| Wanted | 0.489 |
| Collectibles | 0.472 |
| CDs/DVDs/VHS | 0.441 |
| Books | 0.420 |

**Table 1: JS divergence with respect to Wikipedia for Amazon and Craigslist.**

**Extracting Non-commercial Queries**

For the non-commercial queries from Wikipedia, we did not use the toolbar logs; the typical entry point to Wikipedia is a general search engine rather than the search box on the Wikipedia home page. Because of this, we employed the click logs of a search engine to mine non-commercial queries. In particular, we selected all the queries for which a substantial fraction of the clicks led to a Wikipedia entry (specifically, we set this fraction to one-fourth of all clicks for the query).

**Enforcing Separation**

To enforce the separation property, we selected the categories from Amazon and Craigslist with the highest JS divergence with respect to Wikipedia. In Table 1, we show the categories with the highest and lowest JS divergences for Amazon and Craigslist versus Wikipedia. Notice that categories "Books", "DVDs", and "VHS" have low divergence in both commercial sources, which is consistent with our intuition that the queries for these categories (mostly specific book or film titles) are ambiguous, as their vocabulary can be easily confused with the vocabulary of general non-commercial queries. In contrast, the high divergence categories contain words that refer to brand names, models, etc., which are typically not part of the vocabulary of non-commercial queries.

Note that for Craigslist, there is a sharp decrease in divergence from "CDs/DVDs/VHS" to "Collectibles". We used this observation to prune out the lowest divergence queries, namely "Books" and "CDs/DVDs/VHS". A similar argument was used in the Amazon dataset to prune out all the categories related to books, music, and movies. In the next section, we empirically show that removing the queries from these categories leads to a significant improvement in the performance of the resulting classifier.

# 5. EMPIRICAL EVALUATION

In this section, we test the hypothesis that automatically extracted labeled data can be used to build high accuracy classifiers. We show the cost-effectiveness of our approach by comparing the performance of classifiers trained with automatically extracted data to the performance of classifiers trained with manually labeled data. We also compare against the case in which a semi-supervised learning technique (in particular, *self-training* [15]) is used to leverage unlabeled data. The other goal of the evaluation is to validate the importance of enforcing the properties proposed in Section 2. To do so, we study the effect of training sets extracted from different data sources (some satisfying the properties, others not) on the performance of the resulting classifiers.

## 5.1 Experimental Setup

**Classifier**

All experiments use a logistic regression classifier trained using a set of $N$ labeled training examples $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i$ is a feature vector with corresponding label $y_i$. For the commercial intent query classification task, $\mathbf{x}_i$ is a fixed-length feature vector computed from a text query. We considered only binary features that represent the presence/absence of unigrams and bigrams in the text of the query (including special begin/end bigrams). The value $y_i \in \{-1, +1\}$ is a class label encoding membership $(+1)$ or non-membership $(-1)$ of $\mathbf{x}_i$ in the commercial intent class. The logistic regression classifier computes the probability of data point $\mathbf{x}$ belonging to class $+1$ as:

$$p(y = +1|\mathbf{x}, \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{w}^T\mathbf{x} + b)}.$$

We find the optimal $\mathbf{w}, b$ using the Orthant-Wise Limited-Memory Quasi-Newton method, which enables handling large feature spaces and a large number of training points [2].

**Performance Metric**

We define precision and recall as follows. Let $C$ be the set of queries that have true commercial intent (as decided by manual labeling). Also, for a particular threshold $\theta \in [0, 1]$ on the probability output by the logistic regression classifier, let $Z_\theta$ define the set of queries that have probability of having commercial intent greater than $\theta$. Then, we define precision and recall at threshold level $\theta$ as:

$$\text{precision @ } \theta = \frac{|Z_\theta \cap C|}{|Z_\theta|}$$

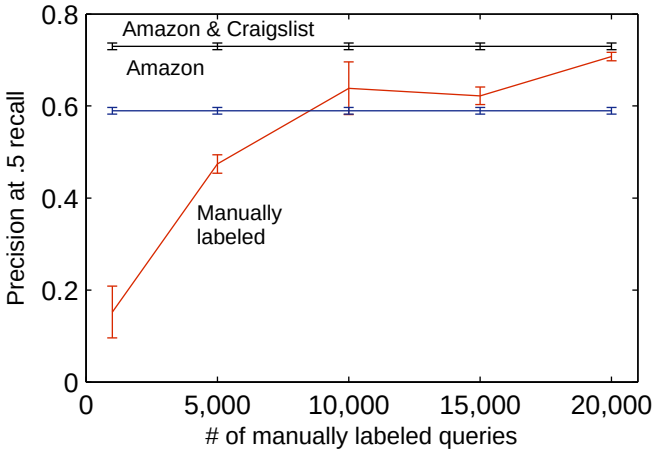


**Figure 2: Comparison of performance of classifiers trained using manually labeled and automatically extracted queries, adding a source at a time.**

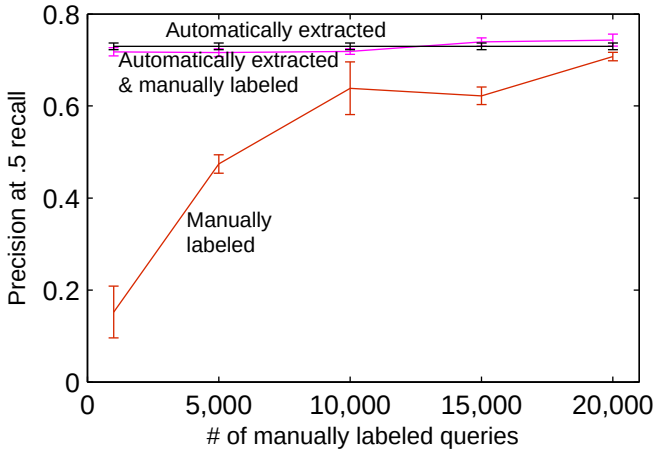


**Figure 3: Comparison of classifier performance in terms of data set size.**

$$\text{recall @ } \theta = \frac{|Z_\theta \cap C|}{|\text{C}|}$$

For many applications of practical importance (such as showing relevant content for commercial intent queries), the goal is to maximize precision for a fixed recall level. Thus, in the following we set the recall to 0.5, which is reasonable for such applications, and we report precision for the value of $\theta$ such that the recall @ $\theta$ is 0.5. We also report the area under the curve (AUC) using a Reimann integral of the entire precision-recall curve.

#### Test Set

The test set consists of 5,000 queries randomly sampled from a search engine query log. In order to assign labels to these queries, we used the Amazon Mechanical Turk Platform.[3] Mechanical Turk is a tool that enables requesters to pose tasks (known as Human Intelligence Tasks or "HITs") to be answered by a community of workers. Since the quality of the labels depends heavily on the design of the HITs, we placed particular emphasis on the HIT design. In particular we asked three questions wherein the "Turker" was asked to weigh plausible commercial and non-commercial intentions for each query and decide the one she thinks is dominant. Each query was shown to five "Turkers," and the final label was assigned to the query using majority voting.

## 5.2 Automatic vs. Manual Labels

The goal of the experiments in this section is to show the cost-effectiveness of using automatically extracted training data, as opposed to manually labeled data. To understand the performance of the classifiers trained with manually labeled data, we constructed training sets by randomly sampling a query log. The labels were obtained using the same process explained above for labeling the queries in the test set, which involves using the Mechanical Turk Platform. We considered training sets of sizes 1K, 5K, 10K, 15K, and 20K. We created six training sets for each size, randomly sampling from a pool of 25K queries, and trained six classifiers for each size of the training set. In the testing phase, we report performance using the mean value and error bars corresponding to one standard deviation.

For the automatically extracted data, we constructed training sets using the procedure of Section 4. We used training sets with queries from Amazon and Wikipedia (roughly 1.5M from each), and training sets that contain queries from Amazon, Craigslist and Wikipedia (1.5M for Amazon and Craigslist, and 3M for Wikipedia). Again, we created six training sets in each case by randomly sampling from queries extracted from the corresponding sources, and we report the mean over the six runs and error bars corresponding to one standard deviation.

Figure 2 shows the performance of the logistic regression classifiers trained with the different types of data. We consider manually labeled training data of different sizes and plot the precision corresponding to a 50% recall in each case. We use horizontal lines to show the performance of the classifiers trained with automatically extracted data. Note that we have used unusually large amounts of manually labeled data to get a feel for the asymptotic region. Such large manual data sets are generally too expensive and time-consuming to construct.

The precision at 50% recall of the classifier trained with Amazon and Wikipedia queries is 0.58. This outperforms the classifiers trained with up to 8.5K manual labels. When we add an additional source of commercial queries (i.e., we use both Amazon and Craigslist), the precision of the classifier trained with automatically generated labels jumps from 0.58 to 0.73. The classifier trained with manually labeled data now needs more than 20K labels to catch up. The same results are obtained in terms of AUC, which is 0.55 for the classifier trained with Amazon, 0.64 for the classifier trained with Amazon and Craigslist, and 0.63 for the classifier trained with 20K manual labels. It is interesting to note what happens for the classifiers trained with the amounts of manually labeled data typically used in practice (1K-5K range). In this case,the performance gap with respect to the classifiers trained with automatically extracted data is significantly large.

[3]https://www.mturk.com/mturk/welcome.

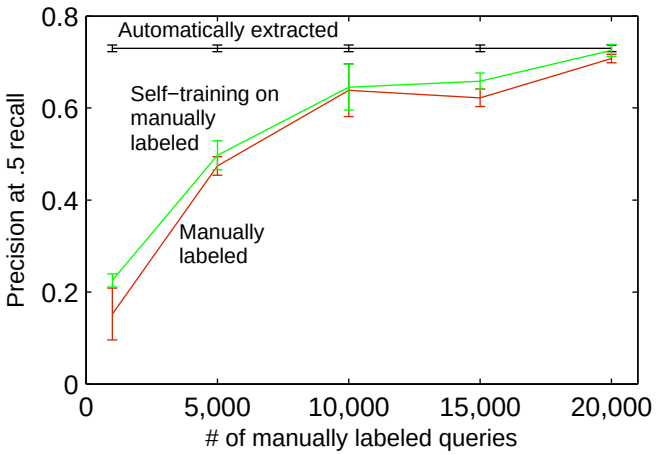


**Figure 4: Performance comparison of classifiers trained using automatically extracted queries and using semi-supervised learning with manually labeled queries.**

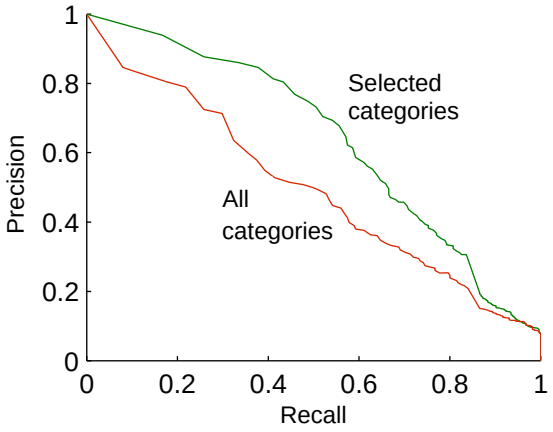


**Figure 5: Precision-recall curves for the classifiers trained with queries from all categories and with queries only from selected categories.**

At this stage, it is natural to wonder whether it is possible to boost the performance of the classifiers trained with automatically extracted data by adding some manually labeled data to the training sets. The answer is yes, but only to a small extent. In particular, we ran an experiment where we consider training sets that consist of a mix of manually labeled queries (from 1K to 25K) and automatically extracted labels. The results are shown in Figure 3. It can be seen that adding manual labels helps to boost the performance of the classifier only when large amounts of labeled data are used; and that even for the largest amount of extra manual labels, the gains are small. In particular, the precision increases from 0.73 to 0.74, which is within the margin of error (the standard deviation is 0.013).

Another natural question is whether it would be possible to boost the performance of the manually labeled training sets by employing semi-supervised learning techniques. To address this question, we considered the case in which we start with classifiers trained with manually labeled data and then apply *self-training*, a semi-supervised learning technique, to exploit unlabeled queries from a query log. In our self-training experiment, we started with the manually labeled queries, and at every round of training, we added the most reliably predicted unlabeled examples to the training set with their putative labels. In particular, for every round of training, we obtained predictions for 5,000 unlabeled queries sampled randomly from the query log of a major search engine. We sorted the probabilities and labeled the top 5% and bottom 10% of the sorted queries as positive and negative examples, respectively. We investigated other schemes, such as hard thresholding of the probabilities, and found the scheme used here to work best. The results are given in Figure 4. We can see that self-training provides only a marginal improvement in precision, and that a large amount of manually labeled data is still necessary to catch up with the classifiers based on automatically extracted data.

## 5.3 Validating Source Properties

In this section, we present experiments aimed at validating the importance of enforcing the properties proposed in Section 2. To validate the separation property, we consider the effect of pruning the low-divergence categories as explained in Section 4. In particular, we consider two classifiers trained with the same number of queries and from the same sources (Amazon, Craigslist, and Wikipedia). The difference between the two training sets is that one contains queries sampled only from the categories that we selected in order to satisfy the separation property; the other contains queries sampled from all categories, including the low divergence ones. In Figure 5, we show the precision-recall curve for both classifiers. It can be observed that our proposed strategy of removing the low divergence categories leads to considerable improvement in precision, especially around recall of 50%.

We also evaluate the importance of the orthogonality property. The reason why we chose Amazon and Craigslist in our study was based on the fact that they belong to orthogonal categories in the Hitwise report: "Department Stores" and "Classified Ads." In order to validate the orthogonality property, we consider what happens when more than one source from the same category is used. In particular, Amazon is the site with the highest traffic for the "Department Stores" category. The second and third most popular sites in the same category are Walmart and Target. We now consider what happens when we use Amazon and one of the other two department stores as sources. We can observe in Figure 6 that adding queries from Walmart or Target does not improve performance with respect to the classifier trained using only queries from Amazon. On the other hand, we can see that when queries from the orthogonal source Craigslist are added, precision improves substantially.

# 6. RELATED WORK

In using automatically extracted training data, we face the problem of potentially noisy data (for example, a query about a commercial product that shares a name with a non-

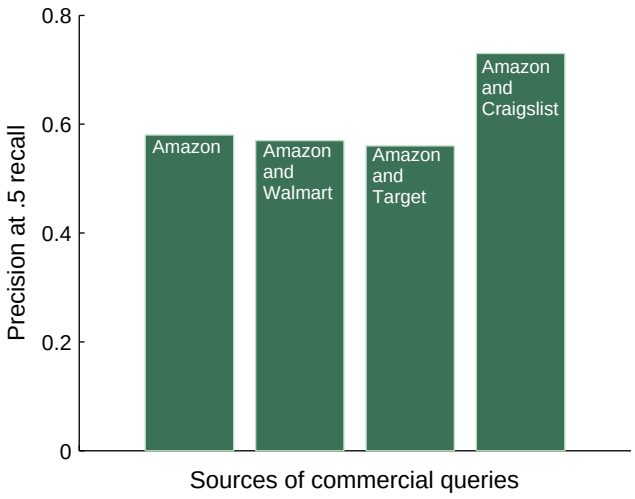


**Figure 6: Precision at 50% recall for various combination of sources for commercial queries.**

commercial object). Previous work on inferring the reliability of training data has relied on measuring similarities between data points. Some approaches assign probabilistic weights to data points based on, for instance, Gaussian mixture models [14] or feedback from a neural network [16]. In [12], the purity of neighborhood graphs (constructed using data features) is used to remove noisy data points. In contrast, we exploit metadata (categories) and background knowledge to reduce the number of undesirable queries.

Most work on query intent identification has used small amounts of labeled data. For example, 6,000 manually labeled queries are used in [3] to learn to classify a query as having informational, navigational, or transactional intent, and only 1,408 labeled queries are used in [7] to train a commercial intent detection classifier. To offset the problems with small training sets, there has been work on using semi-supervised learning. For instance, both [5] and [11] apply semi-supervised learning for the task of web query classification. While [5] uses query logs as unlabeled data, [11] uses the query-click graph. While semi-supervised techniques assume the presence of an initial high-quality seed set of manual labels, our approach, in contrast does not require any manual labeling. Instead, our approach focuses on identifying data sources that can be leveraged to obtain large-scale labeled data for the classification task at hand.

Related, but not directly relevant, is the line of work on mining search engine query logs to obtain training data for learning ranking models [1, 8, 9]. Their task is different and the source of the training data is closely tied to the task at hand. In particular, the click logs collected through the usage of the ranker is utilized to provide implicit feedback to the ranker. In our case the data sources we considered are not directly tied to the classification task.

# 7. CONCLUDING REMARKS

We showed how to leverage good, but imperfect, Web sources to quickly and cheaply generate massive training sets as frequently as needed, in a manner that yields high accuracy classifiers. These techniques may obviate the need for expensive, time-consuming manual training sets for some tasks.

In this paper, we used the task of commercial intent identification to validate our proposal for extracting massive amounts of training data. It is natural to think of other tasks that can benefit from a similar approach. Examples of such tasks include identification of geographical queries where positive queries can arise from popular mapping sites, while negative queries can be the log queries that never (or with very small probability) lead to clicks on mapping sites.

There are several directions for future work. These include exploring other types of sources of labeled data, new techniques for handling noise in the sources, and algorithms to automatically select the thresholds to prune out low-divergence classes.

# 8. REFERENCES

[1] R. Agrawal, A. Halverson, K. Kenthapandi, N. Mishra, and P. Tsaparas. Generating labels from clicks. In *WSDM*, pages 172–181, 2009.

[2] G. Andrew and J. Gao. Scalable training of l1-regularized log-linear models. In *International Conference on Machine Learning*, pages 33–40, 2007.

[3] R. A. Baeza-Yates, L. Calderón-Benavides, and C. N. González-Caro. The intention behind web queries. In *SPIRE*, pages 98–109, 2006.

[4] M. Banko and E. Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the Association for Computational Linguistics*, pages 26–33, 2001.

[5] S. Beitzel, E. Jensen, O. Frieder, and D. Grossman. Automatic web query classification using labeled and unlabeled training data. In *In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 581–582. ACM Press, 2005.

[6] T. Cover and J. Thomas. *Elements of information theory*. Wiley New York, 1991.

[7] H. Dai, Z. Nie, L. Wang, L. Zhao, J. Wen, and Y. Li. Detecting online commercial intention. In *In Proceedings of the 15th International World Wide Web Conference (WWW-06)*, pages 829–837, 2006.

[8] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, pages 133–142. ACM, 2002.

[9] T. Joachims, L. A. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst*, 25(2), 2007.

[10] M. Lapata and F. Keller. Web-based models for natural language processing. *ACM Transactions on Speech and Language Processing*, 1(1):1–31, 2005.

[11] X. Li, Y. Wang, and A. Acero. Learning query intent from regularized click graphs. In *ACM SIGIR conference on Research and Development in Information Retrieval*, pages 339–346, 2008.

[12] F. Muhlenbach, S. Lallich, and D. A. Zighed. Identifying and handling mislabelled instances. *Journal of Intelligent Information Systems*, 22(1):89–109, 2004.

[13] P. Nakov and M. A. Hearst. Using the web as an implicit training set: Application to structural ambiguity resolution. In *HLT/EMNLP*, pages 835–842, 2005.

[14] U. Rebbapragada and C. E. Brodley. Class noise mitigation through instance weighting. In *ECML '07: Proceedings of the 18th European conference on Machine Learning*, pages 708–715, Berlin, Heidelberg, 2007. Springer-Verlag.

[15] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.

[16] X. Zeng and T. R. Martinez. A noise filtering method using neural networks. In *Proceedings of the International Workshop of Soft Computing Techniques in Instrumentation, Measurement and Related Applications*, pages 26–31, 2003.

[17] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.