Generating Succinct Titles for Web URLs

Deepayan Chakrabarti Ravi Kumar Kunal Punera

Yahoo! Research
701 First Ave.
Sunnyvale, CA 94089.
{deepay,ravikumar,kpunera}@yahoo-inc.com

ABSTRACT

How can a search engine automatically provide the best and most appropriate title for a result URL (link-title) so that users will be persuaded to click on the URL? We consider the problem of automatically generating link-titles for URLs and propose a general statistical framework for solving this problem. The framework is based on using information from a diverse collection of sources, each of which can be thought of as contributing one or more candidate link-titles for the URL. It can also incorporate the context in which the link-title will be used, along with constraints on its length. Our framework is applicable to several scenarios: obtaining succinct titles for displaying quicklinks, obtaining titles for URLs that lack a good title, constructing succinct sitemaps, etc. Extensive experiments show that our method is very effective, producing results that are at least 20% better than non-trivial baselines.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation

Keywords

Web page title generation, quicklinks, sitemaps

1. INTRODUCTION

It is well-known that the propensity of a user to click a hyperlink is highly influenced by the anchortext of the link. Both content creators and search engines have constantly exploited this fact to attract more user clicks. Content creators tend to provide hyperlinks with meaningful anchortext to make intra-site navigation convenient for the user; this is heavily used for ranking purposes by web search algorithms.

KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA.

The task of the content creators is somewhat easy since they presumably understand the topology of their own site and can use content-management systems to cope with scale. A search engine, on the other hand, is faced with a more challenging task: even if a web page is recognized to be a perfect result for a user query, the search engine has to automatically provide the right 'title', which we call the *link-title*, and a summary, so that users will be persuaded to click on the search result. Providing an appropriate title is extremely important as eyetracking studies[1] have shown that search engine users focus a lot of their attention on the link-title of the results, even more than the summary provided along with it.

There is a first-cut solution to the above task: use the (HTML) title of the URL itself as its link-title in the search results. This seemingly obvious solution, however, has two major issues. The first issue concerns the availability and quality of titles for URLs. At least 17% of HTML documents lack titles. (We obtained this estimate by analyzing one million random URLs.) Moreover, even if the URL has a valid title, it could be erroneous, incomplete, long, or simply not the best title. For example, at the time of writing this paper, the URL `www.sigir2008.org/schedule.html` on the SIGIR 2008 website has the title 'SIGIR'08 - Singapore' — this is clearly incomplete since 'Conference Schedule,' the topic of this web page, is not indicated in its title. The second issue concerns the presentation and user-experience considerations of search results. The search engine has limited real-estate to present the link-titles of search results and hence they cannot be overly long. The real-estate is even more critical if the URL is displayed as a *quicklink*[2] along with a search result. For example, `www.kdd2007.com/program.html` can be displayed as one of the quicklinks with link-title "Program Information" along with the parent URL `www.kdd2007.com`, for the query "KDD 2007." Here, the search engine must avoid presenting redundant information in the link-title of the quicklink; for example, the link-title "KDD 2007 Conference - Program Information" is arguably superfluous for the above quicklink. The link-title of the quicklink must therefore be derived keeping in consideration the *context* (i.e., the parent URL) in which it is displayed.

In addition to the URL title itself, there are other natural sources of information that can yield potential candidates

[1] such as `www.checkit.nl/pdf/eyetracking_research.pdf`

[2] Quicklinks are typically site entry points that are displayed along with a parent URL, often when the parent URL is the top result of a web query. The main purpose of quicklinks is to provide shortcuts for navigational queries.

for the link-title. For the above quicklink example, these include the following: the tokens in the quicklink URL itself ("program," "kdd2007"), the most distinctive terms on the quicklink web page determined by some criteria (e.g., "Program Information," "General Schedule"), the web search queries for which this URL was clicked by some user (e.g., "kdd 2007 schedule," "2007 kdd program"), the tags, if any, for this URL on `del.icio.us`, the anchortext from the parent URL to the quicklink URL (in this case, it turns out to be the uninformative "here"), the anchortext on hyperlinks from other web pages to the quicklink URL, etc. The question now is: can we automatically combine the information from these disparate sources in a principled way, taking the context and constraints into account, to arrive at the best link-title?

Our contributions. We consider the problem of link-title generation for URLs. Our formulation has two pieces to it. The first is a collection of sources, each of which can contribute one or more candidate link-titles for the URL. The second is the context in which the link-title will be used and constraints on the length of the link-title. These pieces are brought together in a statistical model based on likelihood. The availability of sources that can contribute to the link-titles and the presence of context/constraints make our problem different from the document summarization/title-generation problems considered in traditional IR.

The intuition behind our model is that, compared to the rest of the vocabulary, words from the appropriate link-title and context of a URL are preferentially used to construct all the text (from various sources mentioned above) associated to the URL. We postulate that the probability of a word "generated" by a source of information for the web page is a convex combination of some function of the link-title, the context, and the full vocabulary. A tempting approach at this point would be to obtain the best parameters for all sources by applying the maximum likelihood principle, using a training set of URLs with labeled link-titles. This approach, however, is inadequate since it does not take into account the quantitative and qualitative differences between the various sources. This necessitates the use of source-specific weights in the likelihood framework; we employ a ranking SVM to learn these weights. Our likelihood framework also incorporates length constraints on link-titles.

Our model is quite general and can be applied to a variety of specific link-title generation tasks. A first possible application is to obtain link-titles for quicklinks. Recall that quicklinks occur in the context of a parent URL and their link-titles are constrained to be very short. Hence, this application fully tests all aspects of our framework from the integration of information from various sources to enforcement of constraints coming from application-specific context. Furthermore, as a by-product of the quicklink title generation task, succinct sitemaps can be automatically constructed for a given website. A second application is to automatically obtain a link-title for a web page that is to be shown as a search result. This is especially useful in cases when the title in the HTML content of the web page either is of poor quality or way too long or just doesn't exist. Notice that this application tests aspects of our approach that are different from the previous one, as the context is absent and the length constraint is less demanding. Finally, a third application is to obtain titles for non-HTML documents, especially, for videos, images, and pdf/word documents. Notice that some of the information sources might not be very useful in this case, e.g., unlike web pages, image and video content is not readily interpretable as text to construct link-titles.

We apply our model and techniques to the first two applications mentioned above. We conduct extensive experiments and evaluate our model against various baselines, including some summarization-based IR techniques. In both applications, our approach achieves the best performance, with improvements of at least 20% over non-trivial baselines.

2. RELATED WORK

In the past decade, there has been a lot of interest in the automatic generation of titles and summaries, with applications including news summaries [10], web page classification [11], and summarizing web pages for hand-held devices [4], among others. Most of these consider the document to be summarized as the *only* source of information, while a few try to combine information from multiple sources into one coherent summary. We discuss both of these below.

Document as the only source. There are two general lines of work on generating titles or summaries under this framework. The first is linguistics-based and uses the deep structure of the page content in order to pick important sentences and phrases, which are then combined to form summaries or titles [5]. The second is based on statistical translation techniques, and uses probabilistic model-based methods to pick relevant titles. Among the methods used are word and n-gram probabilities, relevance scores, or a singular value or HITS decomposition of the sentence-word matrix [7, 3, 8, 14]. Extra information such as web page clickthrough rates are also used, but only to modify the relevance scores for words and phrases already present in the text [12].

While many of these approaches perform well under certain settings, there are several reasons why they are not relevant to link-title generation. First, by looking at the document in isolation, it is not possible to incorporate rich sources of side information that are often available for web pages, such as anchortext of inlinks to the web pages, or user tags placed on those. Indeed, in our experiments, the content of a given web page is found to be inferior as a source of link-title information than most others. Second, none of the prior work generates titles in the context of another document or web page, which is critical for applications such as quicklinks and sitemaps. Finally, algorithms that depend on computing probabilities are often aimed at particular domains, and, as our experiments show, often do not scale well when applied to a corpus as large and varied as the web.

Combining multiple sources. The problem of combining multiple sources of information for title or summary generation seems to be relatively less well studied. Radev et al. [10] generate a summary of a news incident by combining information about the same incident from multiple online news sources. Their method uses template operators that can be used to search for contradiction, refinement, agreement, and other such descriptors of the relations between pairs of sources. These templates are then used to output a combined summary of the news incident. Wang et al. [13] combine multiple data sources by an approach based on latent semantic analysis, but do not apply it to title generation. Goldstein et al. [6] propose the maximum marginal relevance

heuristic to generate a query-dependent summary by adding sentences that are both relevant to the query and the document, while having minimal similarity to sentences already in the summary.

All of these methods focus on building a summary that is a combination of sentences (or phrases) from multiple sources. However, this is not very relevant for us: a link-title must be succinct and present one idea, not a combination of words or phrases with possibly different semantics. Also, combining sentences in this fashion is not the same as generating a link-title (or summary) under a given context; in fact, sentences that are already known from the context should in fact be *excluded* from the summary. The methods outlined above do not do this.

Hence, we can see that the necessity of combining sources while under the constraints of a given context leads to novel problems, which are not fully addressed by prior work.

3. PROPOSED METHOD

There are three points that set our problem apart from traditional title generation problems. First, we want to use information contained not just in a single document (web page) but also in other relevant sources — queries for which the web page was viewed or clicked, `del.icio.us` tags, the URL of the web page, hyperlinks to the web page, and so on. The relevance of each source to the link-title generation process might be different, and this must be accounted for. Second, the link-title must be generated under a certain context: the link-title for a quicklink `fedex.com/Tracking` for the site `fedex.com` must focus more on the "tracking" aspects of the quicklink and less on generic "Fedex" aspects available from the main website page. More generally, the link-title of a given web page must be constructed to emphasize aspects that *differentiate* it from the context provided by another given web page. Finally, there is an extreme skew towards short link-titles, with user studies showing single-word link-titles to be the most preferable. This encourages the use of extraction-based link-title generation, since it is highly likely that a word or short phrase already existing in a source, after small modifications, will be an excellent link-title. Indeed, this is the approach we propose.

Formally, we have available a set $\mathcal{S}$ of sources of information for each web page. Associated with every pair (w, s) of a web page w and a source $s \in \mathcal{S}$ is a (possibly empty) set $\mathcal{I}(w, s) = \{(t_1, x_1), \ldots, (t_n, x_m)\}$, where each tuple (t_i, x_i) represents a text instance and its corresponding weight, respectively. For example, the CLICKED-QUERIES source for the web page `fedex.com/Tracking` may contain the tuple ("Fedex Tracking Number", 51), with the first field being a search engine query for which `fedex.com/Tracking` was returned as a result and clicked, and the second field the number of such occurrences. Let $\mathcal{S}_c \subset \mathcal{S}$ be the set of sources suitable to extract candidates link-titles from; soon, we will discuss how to choose this subset $\mathcal{S}_c$. Then we denote by $\mathcal{I}(w) = \cup_{s \in \mathcal{S}} \mathcal{I}(w, s)$ and $\mathcal{I}_c(w) = \cup_{s \in \mathcal{S}_c} \mathcal{I}(w, s)$ the set of all text instances and link-title candidates respectively for web page w. Slightly abusing notation, we will also use $\mathcal{I}(w)$ and $\mathcal{I}_c(w)$ to refer to just the texts (i.e., the first fields), ignoring the weights. Now, the problem is defined as follows:

PROBLEM 1 (CONTEXT-BASED LINK-TITLE SELECTION). *Given a context web page b and a specific web page w, along with $\mathcal{I}(b)$, $\mathcal{I}(w)$, and a candidate link-title set $\mathcal{I}_c(w)$, pick the best link-title $T(w, b) \in \mathcal{I}_c(w)$ for w with respect to b.*

Note that $T(w, b)$ need not be the same as the contents of the `<title>` field in the HTML of web page w; indeed, the latter need not even exist.

Our solution uses extraction methods to generate candidate link-titles, which are then ranked using statistical methods, and the highest-ranked candidate is returned as the link-title. In the following, we describe the sources and the generation of candidate link-titles, and then present the statistical model used for the task.

3.1 Sources of text instances and link-titles

There are many different sources of information regarding any given web page. These include the URL, title, and key phrases [1, 2] of the web page, anchortext on links pointing into the web page, search queries for which the web page was returned as a top result, and any user-generated tags for that web page. The full list of sources we use is described in Table 1.

Source	Description
INTRA-AT*	Anchortext on intra-site links
INTER-AT*	Anchortext on inter-site links
AT-FROM-HP*	Anchortext on link from b to w
VIEWED-QUERIES	Search queries for which w was returned in the top 10 results
CLICKED-QUERIES*	Search queries for which w was returned as a result, and clicked
FIRST-CLICKED-QUERIES*	Search queries for which w was the *first* result, and clicked
PAGE-TITLE*	Title of w
URL-TOKENS	Word tokens from the URL of w
PRISMA*	Key phrases in w's content extracted by [1, 2]
DELICIOUS	Tags for w from `del.icio.us`

Table 1: Sources of text instances and candidate link-titles: the specific web page under consideration is w, and the context web page b, and the starred sources belong to $\mathcal{S}_c$.

Since link-titles are typically very short, the odds of an existing word or phrase from these sources being the link-title are high. However, not all sources are good for link-titles. Spelling mistakes may be common in some sources (e.g., queries that don't generate clicks), while some might not even be complete phrases (e.g., token from the web page URL). Thus, we only use a subset of the sources for candidate link-titles, and these are starred in Table 1.

3.2 Statistical model

The intuition behind our model is that, compared to the rest of the vocabulary, words from the link-title $T(w, b)$ and the context web page instances $\mathcal{I}(b)$ are preferentially used in all the text instances $\mathcal{I}(w)$ associated with web page w. However, the degree of preference may depend on whether the word occurs in the link-title, or in the context $\mathcal{I}(b)$, or both. In addition, not all sources are created equal: the source INTRA-AT might use many more words from the link-title than, say, the source URL-TOKENS. In fact, the latter source is more likely to use words associated with the context web page $\mathcal{I}(b)$. The model must thus differentiate between the sources as well.

Formally, in the generative model we associate with each source $s \in \mathcal{S}$ two parameters α_s and β_s, with the following semantics. Whenever a new word needs to be "generated" by source s for web page w, it is drawn from the words in $T(w,b)$ with probability α_s, from $\mathcal{I}(b)$ with probability β_s, and from the full vocabulary V with the remaining probability. In practice, we slightly modify this formulation by replacing $\mathcal{I}(b)$ above with a specially chosen subset $W(b)$, described in Section 4.1, which improves performance. Thus, the probability of generating word x from source s is given by

$$P_s(x \mid W(b), T(w,b)) = \alpha_s \cdot \frac{\#\{x \in T(w,b)\}}{|T(w,b)|} + \beta_s \cdot \frac{\#\{x \in W(b)\}}{|W(b)|} + (1 - \alpha_s - \beta_s) \cdot \frac{\#\{x \in V\}}{|V|}, \quad (1)$$

where the $\#\{\cdot\}$ notation denotes the number of times x occurs in a given multiset, and $|T(w,b)|$, $|W(b)|$, and $|V|$ represent their sizes[3].

Equation 1 ties the data observations (i.e., the words generated by the sources) with the link-title of the page and the source parameters α_s and β_s, and thus allows us to both (1) *fit* the model parameters when provided the correct link-title $T(w,b)$ (i.e., the training phase), and then (2) *infer* the best link-title for new (w,b) pairs using known model parameters (i.e., the testing phase). However, the quality of results is strongly dependent on the details of this process. We next look at two formulations for training and testing. The first, a naive formulation, serves to illustrate the basic ideas. These are built upon by the second formulation, which makes the model more realistic and accurate, but at the cost of increased complexity in the model fitting process.

Naive formulation. Suppose we know α_s and β_s for all sources s. Now, given w and b (and hence $W(b)$), we could naively compute the likelihood of any candidate link-title t:

$$L(t \mid w, b, W(b)) = P(\mathcal{I}(w) | W(b), t) = \left(\Pi_{s \in \mathcal{S}} \Pi_{(x,n) \in \mathcal{I}(w,s)} \left[P_s(x \mid W(b), t)\right]^n\right) \cdot P_{\text{len}}(|t|), \quad (2)$$

where $P_{\text{len}}(|t|)$ is the *a priori* probability of the link-title being a certain length, and can be easily determined from a training set. Note that this formulation assumes that the sources are independent, which is clearly untrue for some sources (e.g., VIEWED-QUERIES and CLICKED-QUERIES), but it serves as a reasonable starting point.

Let us consider qualitatively the effect of this formula. Suppose a word x occurs repeatedly in $\mathcal{I}(w)$. The corresponding $P_s(x)$ terms will significantly affect the likelihood (Equation 2), whose maximization will in turn require higher values of $P_s(x)$. This happens if x occurs in the candidate link-title t. Thus, link-titles containing frequently occurring words are preferred, as expected. However, there is also a strong source-specific dependence. Suppose a source s is highly likely to use words from the link-title, i.e., $\alpha_s \approx 1$. Then, any candidate link-title t that does not include a word x from s will cause extremely low $P_s(x)$ values, dragging down the likelihood and reducing the candidate's appeal. The presence of the β_s term is also critical: had α_s been the only parameter, then *any* repeated words, even those that occur frequently in the context web page, would be preferentially picked to be in the link-title. But now, the β_s term ensures that such words have relatively high $P_s(x)$ values even if they do not occur in the link-title; the relative increase in $P_s(x)$ (and the likelihood) if we added these terms to the link-title is much less, thus reducing the pressure to have these terms in the link-title. In fact, the pressure from P_{len} to have short titles, especially for the quicklinks title task, will decrease the chance of words from $W(b)$ being present in the link-title $T(w,b)$.

Training phase. Parameter fitting is simple under the naive model. Given a training set of web pages, context pages, and their true link-titles, we can fit α_s and β_s for all sources by maximizing the likelihood function (2) with respect to these parameters. For this, we first write down the log-likelihood function

$$\ell(t \mid w, b, W(b)) = \left(\sum_{s \in \mathcal{S}} \sum_{(x,n) \in \mathcal{I}(w,s)} n \cdot \log P_s(x \mid W(b), t)\right) + \log P_{\text{len}}(|t|), \quad (3)$$

and then finding the parameter values where its derivative goes to zero. Note that the α_s and β_s parameters for different sources "factor out," i.e., there are no terms in the log-likelihood that include parameters from two different sources. This factoring of the log-likelihood implies that the parameters for each source can be optimized independently of other sources, thus further simplifying the parameter estimation process. Only one sequential pass over the training data is needed for this computation.

Testing phase. Given a set of candidate link-titles, we compute the likelihood of each candidate and return the one with the highest likelihood as the result.

Full formulation. While the simplicity of the naive formulation is very appealing, it suffers from two problems. First, there might be imbalances in the number of instances $|\mathcal{I}(w,s)|$ for the different sources $s \in S$. For instance, the CLICKED-QUERIES source could consist of many different query instances, while the URL-TOKENS source yields only one instance, that of the URL of w broken up into tokens (e.g., "Music India Online" for `www.musicindiaonline.com`). Since the naive formulation counts each instance equally, sources with few instances can get swamped and have their importance reduced, even if they are good predictors of the correct link-title. Second, even if all sources could be normalized to have the same number of instances, the instances of some sources are still "noisier" than others. Consider, for example, the CLICKED-QUERIES and the VIEWED-QUERIES sources. Instances of the VIEWED-QUERIES source are search queries for which web page w was returned as a result by the search engine; for CLICKED-QUERIES, the search result for web page w was also *clicked* by the user. Thus, CLICKED-QUERIES are expected to have less noise than VIEWED-QUERIES, and a full formulation should account for such differences between sources. Finally, the log-likelihood for the naive model (Equation 4) assumes independent sources, which need not be true in general.

Our approach to this problem is to apply a source-specific normalization to the instances. In particular, every instance $(x,n) \in \mathcal{I}(w,s)$ of source s is given a weight $\theta_s / |\mathcal{I}(w,s)|$, where θ_s is a source-specific parameter and $|\mathcal{I}(w,s)|$ the total number of instances for source s. This can also be thought of as building a histogram over all the words generated by

[3] $T(w,b)$, $W(b)$, and V are considered as bags-of-words for this.

the source, and then normalizing the histogram so that it sums up to θ_s. The new log-likelihood function is:

$$\ell(t \mid w, b, W(b)) = \sum_{s \in \mathcal{S}} \theta_s \cdot \sum_{(x,n) \in \mathcal{I}(w,s)} \left(\frac{n}{|\mathcal{I}(w,s)|} \cdot \log P_s(x \mid W(b), t) \right) + \theta_{\text{len}} \cdot \log P_{\text{len}}(|t|) \quad (4)$$

Note that the addition of the θ_s parameters allows the sources to be dependent: for example, if two sources are identical, a good training algorithm will learn $\theta_s \approx 0$ for one of these two sources.

Training phase. Under this formulation, we must estimate not only α_s and β_s, but also θ_s for each source s. The first two can be learned as in the naive formulation discussed above, but learning θ_s presents some unique challenges. If we merely attempt to find the θ_s values that maximize the log-likelihood, then some θ_s parameters can grow to unbounded magnitude. Constraining the $(\theta_1, \ldots, \theta_{|S|})$ vector to lie within a unit ball, in any L_p-norm, leads to a solution where one θ_s value is one, and all the rest zero[4]. Clearly, neither of these is acceptable, and a different approach is needed for fitting θ_s.

Our solution is to learn θ_s using *extra* information that is unavailable in Equation 4. Note that up to this point, only the correct link-title has been used for training; now we also tell the learning routine about the quality of the available candidate link-titles. The obvious approach would be to compute the similarity between a candidate link-title and the correct link-title, and to learn the θ_s parameters by linear regression to these similarity values. However, this approach has some pitfalls. Imagine two different web pages w_1 and w_2 with identical instance sets $\mathcal{I}(w_1) = \mathcal{I}(w_2)$. However, they might have completely different link-titles, due to differences in the wording of the correct link-titles, or differences in the precise content of w_1 and w_2 that is too fine-grained to be picked up by the available sources, or any other such factors. The similarity of any given candidate link-title to the correct link-titles would be completely different for the two web pages, making the regression problem undefined. In general, it is not the exact similarity *value* that is important, but rather the *rankings* of the different candidates. In fact, we observe empirically that the rankings remain almost identical for several different similarity functions, including Jaccard similarity, precision, and f-measure. Thus, the ranking of candidates is a better base to learn from, as compared to the similarity values themselves.

The availability of such training data in the form of rankings suggests the use of a learning algorithm based on pairwise preferences, such as Ranking SVM [9]. Indeed, for known values of α_s and β_s, Equation 4 becomes a linear function in the θ parameters. Thus, we use the following two-step approach to fit the parameters under this formulation: (1) fit α_s and β_s separately for each source s, by maximizing the log-likelihood and using its "factoring" property, and then (2) learn the θ_s values (and similarly, θ_{len}) using a linear ranking SVM. Empirical results described later show how this negative information present in the rankings can help improve accuracy of the model, as well as significantly shorten training times. The learned values of θ_s are also indicators of the relative importance of the various sources, and can aid in interpreting the final model.

[4]This follows from the fact that the coefficients of the θ_s terms and θ_{len} are sums of log-probabilities, which are all non-positive.

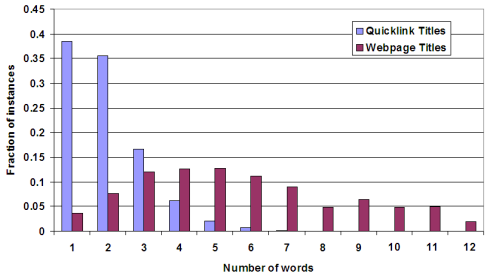

Figure 1: Distribution of title lengths for quicklinks, and for web pages without any context web page. Clearly, the two are fundamentally different.

Testing phase. This is identical to that in the naive formulation: using Equation 4, we compute the log-likelihood of each candidate link-title, and then pick the best among them as our proposed link-title.

4. EMPIRICAL EVALUATION

In this section, we present an empirical evaluation of the performance of our approach on a host of real-world scenarios. We constructed datasets for the task of predicting titles for web pages both within and without the context of another web page. In order to put the performance of our approach in perspective, we measure the performance of many intuitive baselines as well as competing approaches like PRISMA [2] and the method of Banko et al. [3], henceforth, BMW. We show that our approach significantly outperforms all baselines and competing approaches. The effects of various factors on our method's performance in these real-world applications are also examined thoroughly. We begin by discussing our experimental methodology.

4.1 Experimental methodology

Here we describe the datasets, evaluation measures, and baselines employed in evaluating our approach on the tasks of predicting link-titles.

Datasets: Ground truth and sources. We created two datasets to empirically evaluate the performance of our approach under real-life scenarios.

Quicklinks titles. This dataset was constructed to simulate the task of predicting titles for web pages within the context of another web page, specifically the website's homepage. Thus, it is useful for performance evaluation on the task of predicting titles of quicklinks, as described earlier in the paper. In order to construct this dataset, we picked a set of around 4K most accessed websites from search engine logs. For these websites, a quicklink selection algorithm picked salient URLs that people often navigate to. These URLs were then shown to three human judges who manually constructed titles that suitably addressed the content of the URLs in the context of the homepage. In this manner 2187 unique titles were constructed for 1430 URLs. Some URLs were labeled with multiple titles when the judges considered it necessary. As we shall describe later, this fact also affects

	Judge1	Judge2	Judge3	Judge1	Judge2	Judge3	Judge1	Judge2	Judge3
Judge1	1			1			1		
Judge2	0.83	1		0.78	1		0.75	1	
Judge3	0.75	0.86	1	0.67	0.77	1	0.48	0.72	1

Table 2: Inter-judge agreement on quicklink titles in terms of f-measure, Jaccard measure, and exact match.

the way we employ our evaluation measures. Table 2 present the inter-judge agreement scores in terms of the evaluation measures we use in this paper. We discuss the impact of these scores later in this section.

Before proceeding any further, we briefly comment on bias in the data. While the websites present in the dataset were picked randomly, the quicklink URLs for the title generation task were picked in a systematic fashion. Hence, the URLs labeled in this dataset are biased towards frequently navigated web pages within the website. However, this bias is necessary to effectively evaluate approaches that construct titles for quicklinks.

Web page titles. As we discussed in Section 1, the title of the web page that is specified in the HTML is often not suitable when the web page is surfaced in a search results page. Under this scenario we want our approach to predict the title which can be used as the link text on the search results page. To simulate this scenario we constructed a dataset of around 60K web pages with known titles, and learn a model that predicts the original title given to the web page by its creator.

As with the quicklink titles dataset, we picked some web pages that were likely to show up in the top results of the search engine. We noticed that ∼17% of these web pages had unusable titles. These web pages were thrown out and the HTML titles of the rest were used as ground truth.

Sources of text instances, vocabulary, and $W(b)$. For the URLs in each of the above datasets, we fetched the various sources of information that are used by our approach. These sources of text instances are described in Table 1. The texts from the sources were processed via porter stemming. Stopwords were retained while processing candidate link-titles for legibility reasons, but were not considered in likelihood computations. The term frequencies in the vocabulary V were computed by processing a large random sample of web pages. The subset $W(b)$ that comprises the words in the context was constructed by taking the top three most common text instances of each source from the context $\mathcal{I}(b)$.

Baselines & competing approaches. In order to place the performance of our approach in perspective, we examine the performance of several baselines as well as published work from the literature. In particular we compare our approach against Prisma [2] and BMW [3].

Prisma. This system was proposed by Anick and Tipirneni [1] for the task of summarizing the contents of a web page or web search results page in as few phrases as possible, so as to provide the user with a succinct description of the content. The Prisma system uses various cues derived from the HTML structure of the web page in order to rank phrases in terms of salience. For instance, phrases within `<h1>` tags, those at the beginning of the web page, and those in bold are ranked higher. We adapt this approach by picking the highest scoring phrase as the predicted title.

BMW. This approach was proposed by Banko et al. [3] (we call it BMW after the author names) and offers an interesting counterpoint to our approach to predicting titles. The BMW approach learns parameters which model the tendency of words and bigrams that occur in the content of a web page to also occur in its title. This is in contrast to our approach where we only learn parameters for word *matches* between different sources of information. In our experience, on a corpus as large and diverse as the web, learning on a per-word level entails a lot of training data and time. As we shall see later, this particular aspect of BMW hampers its accuracy in generating titles. The bigram probabilities in this approach, while ensuring that generated titles are usually grammatically correct, nonetheless increase the amount of data and time needed for training even more. In fact, in our evaluation the bigram based model took an inordinate amount of time to train and had very poor accuracy (because of sparsity of data). Hence, we only report results derived after turning off the bigram probabilities.

Baselines. While BMW and Prisma constructs titles for web pages in general, we didn't find any work in the literature that constructs titles for the constrained case of quicklinks. Hence, in our evaluation, we juxtapose the accuracy of our approach against the predictive accuracy of the various sources of information that it relies on. Hence, the baselines in our evaluation are predicting the candidates from various sources as titles of quicklinks. As we shall show later in the section, these are surprisingly strong baselines. We will analyze their accuracy and present reasons for their success. We will also show that our approach picks the best candidate from the various sources, leading to much better performance than any one source.

Evaluation measures. Evaluation of titles is a challenging problem since we need to determine both whether the predicted titles are coherent and whether they represent the ideas central to the web page. To ensure that we obtain coherent titles we make sure to never change any candidate title obtained from individual sources, each of which is assumed to be coherently constructed. Also, we don't use sources like URL-Tokens and Delicious as candidates titles, since they tend to less "title-like." The list of sources $\mathcal{S}_c$ which are used as candidate titles is given in Table 1. In order to evaluate the generated titles in terms of similarity to true title, we use the following standard measures.

F-measure. We define the precision of a predicted title as the number of words in it that also occur in the true title, and the recall as the number of true title words that occur in the predicted title. F-measure is the harmonic mean of these two quantities and measures how well the predicted title and true title agree. A higher value of f-measure indicates greater agreement.

Jaccard measure. We can also measure the degree of overlap between the predicted and true titles using the Jaccard

Approach	F-measure	Jaccard	Exact match
Our approach	**0.81**	**0.75**	**0.63**
AT-FROM-HP	0.70	0.66	0.58
INTRA-AT	0.43	0.41	0.35
INTER-AT	0.36	0.32	0.25
PAGE-TITLE	0.37	0.27	0.05
CLICKED-QUERIES	0.25	0.19	0.07
PRISMA	0.24	0.22	0.13

Table 3: Performance of various approaches on the task of predicting titles for quicklinks.

measure. If we regard both titles as sets of words, then the standard Jaccard measure is defined as the ratio of the size of intersection to the size of union of the two sets. In particular, we use a multi-set version of the Jaccard measure. This is computed as $\frac{\sum_w \min(P(w),T(w))}{\sum_w \max(P(w),T(w))}$, where w iterates over words, and $P(w)$ and $T(w)$ are the number of times w occurs in the predicted and true title respectively. This measure has the effect of penalizing unnecessarily repeated words in the predicted title as this can sometimes lead to diminished user experience.

Exact match. Both measures mentioned above compute accuracy independent of the word ordering. However, we would also like that the predicted title be coherently worded, and not just a random permutation of useful words. In order to evaluate our approach using this, we compute the fraction of test instances for which it predicts the exact true title.

Longest common subsequence (LCS). The exact match measure is more meaningful for the task of site-map title prediction than web page title prediction. This is because true web page titles tend to be longer and hence in our evaluation almost all approaches score zero in the exact match criteria. Hence, for the web page title generation task we evaluate our approach based on the length of the longest common subsequence of words between the predicted and true titles.

Difficulty of evaluating semantic similarity. Approximating the semantic similarity of predicted titles to true titles via the syntactic measures mentioned above is challenging because in natural language the same ideas can be expressed in many ways. This is not so much an issue with quicklink titles since they are smaller and more specific. Still, we hope that we can use these measures for relative comparison of different approaches since each approach will be impacted by these issues equally. However, the absolute numbers reported with these measures can be regarded as lower bounds on the actual performance of our approach.

To give a sense of the inherent variability of ground truth, in Table 2 we present the inter-judge agreements computed based on double labeling of quicklinks titles. The numbers are reported in terms of the measures that we use to evaluate our algorithms. Hence, one judge is considered to be ground truth, which the other judge is evaluated against: all measures we consider are symmetric. As we can see, judges agree with each other to a significant extent, indicating that a learning based approach should work. However, the agreements are not perfect, indicating that these numbers should serve as an approximate upper-bound on how well we can expect the best possible algorithms to perform.

4.2 Results on quicklinks titles task

Here we report our results on the task of predicting quicklink titles. Various sources of information used by our approach serve as baselines. The title generation approach BMW does not consider context while predicting quicklink titles and hence is not competitive on this particular task: we do not include it in the evaluation in this section.

Comparison with baselines. In this section we analyze the performance of our approach and compare it against several baselines. The performance numbers are presented in Table 3. The first thing to note is that our approach performs extremely well. In fact, the scores obtained by our approach are very close to those presented in the inter-judge agreement (Table 2), indicating that significant further improvement is not possible given the variability of the task. Moreover, in all three measures, our approach far outperforms all other baselines.

Now let us consider the performance of baselines that our approach uses in the task of predicting quicklink titles. The most logical baseline is the HTML title of the web page represented by the PAGE-TITLE source. However, as we see in Table 3, the PAGE-TITLE source does not serve as a good quicklink title. This is primarily because PAGE-TITLE is often used to incorporate information about the website, sometimes for branding purposes. Hence, the candidates in the PAGE-TITLE source are not constructed in the context of the homepage, resulting in poor precision. Another reason is the difference in lengths of the quicklinks titles as labeled by human judges and the titles of the web pages (Figure 1). A second source that uses the text within a web page to predict the quicklink title is PRISMA. The PRISMA approach uses multiple syntactic cues from the HTML code of the web page (like text within `<h1>` tags, bold text, etc.) to extract candidates that best describe the web page. However, because the information present on a web page can be diverse and noisy, the title candidates predicted by PRISMA score low on all the measures (Table 3).

Sources based on anchor-text produce candidates that score very well as quicklink titles. This is because they are often created by humans trying to describe the target web page in a succinct manner. Even within the class of anchor-text based sources the INTRA-AT source outperforms the INTER-AT source. This is because when people construct the anchor-text for intra-site links they are working to describe the target page in the context of the website. An extreme example of this is the anchor-text on the homepage of the website. In this particular case, all the information about the website is present on the homepage, and hence, none of it has to be present in the anchor-text. Consequently, as we can see from Table 3, the AT-FROM-HP source scores extremely well as a predictor of quicklink titles.

Thus, the results in Table 3 demonstrate that our approach effectively selects the best candidate from the various individual sources of information about the web page.

Learning rates. Next, we examine the amount of labeled data that is needed by our approach for training the models. In Figure 2 we plot the accuracy of the quicklink titles predicted by our approach against the amount of training data used to learn the models. As we can see, even with as few as 15 labeled quicklinks, the performance of our approach is better than always using the AT-FROM-HP source to predict

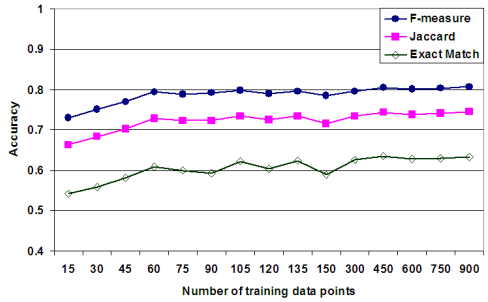

Figure 2: Learning rate plots of our approach on the quicklink title prediction task. As we can see, the accuracy of our approach rises rapidly and stabilizes after processing a very small number of labeled datapoints.

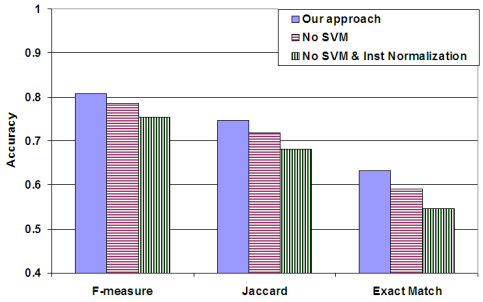

Figure 3: The performance of our approach without any source specific learning and without instance normalization on the quicklink title prediction task. Both enhancements to our approach serve to increase the quality of predicted titles.

quicklink titles. This shows that our approach uses very few datapoints to quickly learn to determine when to predict the AT-FROM-HP as the true title and when to use some other appropriate source. Moreover, as we increase the amount of supervision given to our algorithm, the accuracy rises rapidly and then stabilizes after around 60 labeled quicklinks have been processed. Therefore, after seeing very few labeled examples, our approach learns to predict quicklink titles with an accuracy that approaches the upper-bound suggested by the inter-judge agreement in Table 2.

Comparison with naive approach. In this section we compare our full approach against the naive model which doesn't learn the weights for sources using the Ranking SVM method and doesn't do any source-specific normalization. The performance of our approach in terms of the three measures is plotted in Figure 3. As we can see both additions to the naive model help increase the quality of the quicklink titles predicted. In order to show what is happening in more detail we show a learning rate graph in Figure 4 that plots the accuracy (in terms of Jaccard) of our full approach and our approach without learning of source weights. As we can see, while our full approach has higher accuracy than not using source specific weights (all source weights are 1), in the initial part of the plot, under very "low-data" conditions, the default source weights produce more accurate models.

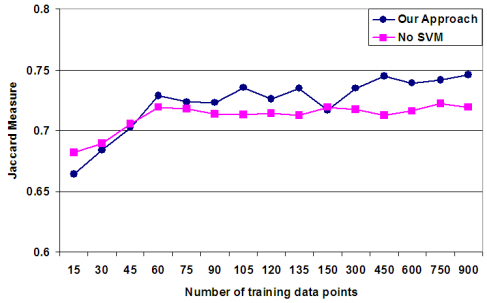

Figure 4: Learning rate plot contrasts our full approach against our approach with no learning of source specific weights.

Approach	F-measure	Jaccard	LCS (words)
Our approach	**0.53**	**0.41**	**3.44**
PRISMA	0.41	0.31	2.54
BMW	0.12	0.10	0.46
AT-FROM-HP	0.45	0.34	2.7
CLICKED-QUERIES	0.31	0.23	2.1
INTER-AT	0.29	0.21	1.8
INTRA-AT	0.28	0.21	1.7

Table 4: Performance of various approaches on the task of predicting web page titles.

This is because when our approach has access to very limited amounts of data the parameters learned by the Ranking SVM method do not generalize well. In this scenario setting all weights equal to 1 as motivated by the MLE framework performs better. However, as we increase the amount of training data, the only increase in the performance of the naive approach is because of better estimates of the α_s and β_s parameters. Hence, as more data becomes available, our full approach that learns source specific weights starts performing better.

Learned parameter values. Recall that the θ_s parameters are indicators of the relative importance of the various sources. This allows us to rank our sources. For the quicklinks titles task, the ranking of sources, in decreasing order of importance, is: AT-FROM-HP, PAGE-TITLE, INTER-AT, URL-TOKENS, FIRST-CLICKED-QUERIES, INTRA-AT, CLICKED-QUERIES, DELICIOUS, VIEWED-QUERIES, PRISMA.

We can make three observations from this list. First, as we would intuitively expect, AT-FROM-HP is the most important source, since this text is provided by the website creator specifically for the purpose of describing the web page w from the context web page b. Second, note that the key phrases obtained from the content of the web page (PRISMA) are the least important. This is because while such phrases are definitely relevant to the quicklink title, they contain a lot of irrelevant information as well; the other sources are much more succinct and relevant to the quicklink title and hence get higher importance. Finally, note that CLICKED-QUERIES are more important than VIEWED-QUERIES. This again agrees with the intuition that user clicks imply increased relevance and should make a source more important.

4.3 Results on web page title prediction task

In this section we report on our results on the task of predicting titles for web pages.

Comparison with competing algorithms and baselines. Table 4 summarizes the performance of our approach and various competing algorithms as well as baselines. In the task of predicting web page titles, the accuracy in terms of exact match was zero for most approaches, hence here we report numbers in terms of the length of the LCS of words found between the predicted and true title. As we see, our approach outperforms all other competing algorithms and baselines by a significant margin in all three measures. This shows that our approach does an effective job of combining various data sources (which are listed in the table as baselines), and predicting an accurate web page title.

Next, let us consider the performance of the competing approaches. Neither PRISMA nor BMW performs as well as our approach in predicting titles of web pages. PRISMA was originally proposed as a web page summarization tool which was adapted here for predicting titles, and hence it has mediocre accuracy for this particular task. However, BMW is a title prediction algorithm which has been shown to have good accuracy on the Reuters dataset. Hence, examining why it fails here provides an interesting perspective into the characteristics that are specific to the task of title prediction for pages on the web.

BMW learns parameters which model the tendency of each word and bigram occurring in the title of a web page. This results in an enormous number of parameters which need a large amount of data from the same domain for robust estimation. While this is possible on a restricted domain like the news stories in the Reuters collection, its extremely difficult on a diverse corpus like the web. Moreover, the large corpus has to be iterated upon and the large number of parameters have to be stored in disk resident fashion, drastically reducing the efficiency of this approach. In fact, the accuracy of the full BMW model was extremely bad, and we had to turn off the bigram parameter estimation to obtain the accuracy reported in Table 4. Finally, with the bigrams turned off there is nothing forcing the correct ordering of words in the title, and hence, as we can see in the results, BMW has a very low score in terms of the LCS measure.

Our approach avoids the parameter explosion inherent in many algorithms applied to the web corpus by learning parameters for the matches between sources, instead of learning parameters for each possible word, bigram, or phrase. This results in few parameters, and hence, robust generalization. Moreover, the relatively few parameters that need to be estimated make our approach fast. Finally, our approach avoids predicting malformed sentences as titles by not changing the candidates obtained from the sources. As Table 4 shows, our approach on average shares almost 3.5 words in the correct order with the true web page titles.

5. CONCLUSIONS

In this paper we considered the problem of automatically generating succinct link-titles for URLs. Our solution to this problem is based on a general statistical framework. The main idea behind our framework is to aggregate information from a collection of sources, each of which can be thought of as contributing one or more candidate link-titles for the URL. Our framework also takes into account the context in which the link-title will be used, and the constraints on its length dictated by real-estate considerations on the search result page. We propose several applications of our framework, including, obtaining succinct titles for displaying quicklinks, obtaining titles for URLs without a good HTML title, constructing succinct sitemaps, and so on. Empirical analysis using manually labeled data shows that our framework is very effective in producing high quality link-titles.

Acknowledgments

We thank Will Chin, Arah Cho, Yong Gao, and Karen Wu for their help with collecting the dataset and evaluating the results. In addition we thank the anonymous reviewers for their valuable comments.

6. REFERENCES

[1] P. Anick. Using terminological feedback for web search refinement: A log-based study. In *26th SIGIR*, pages 88–95, 2003.

[2] P. Anick and S. Tipirneni. The paraphrase search assistant: Terminological feedback for iterative information seeking. In *22nd SIGIR*, pages 153–159, 1999.

[3] M. Banko, V. O. Mittal, and M. J. Witbrock. Headline generation based on statistical translation. In *38th ACL*, pages 318–325, 2000.

[4] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke. Seeing the whole in parts: Text summarization for web browsing on handheld devices. In *10th WWW*, pages 652–662, 2001.

[5] B. Dorr, D. Zajic, and R. Schwartz. Hedge trimmer: A parse-and-trim approach to headline generation. In *HLT-NAACL Text Summarization Workshop*, pages 1–8, 2003.

[6] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. Creating and evaluating multi-document sentence extract summaries. In *9th CIKM*, pages 165–172, 2000.

[7] Y. Gong and X. Liu. Generic text summarization using relevance measure and latent semantic analysis. In *24th SIGIR*, pages 19–25, 2001.

[8] R. Jin. *Statistical Approaches Toward Title Generation*. PhD thesis, Carnegie Mellon University, 2003.

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

[10] D. R. Radev and K. R. McKeown. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):470–500, 1998.

[11] D. Shen, Z. Chen, Q. Yang, H.-J. Zeng, B. Zhang, Y. Lu, and W.-Y. Ma. Web-page classification through summarization. In *27th SIGIR*, pages 242–249, 2004.

[12] J.-T. Sun, D. Shen, H.-J. Zeng, Q. Yang, Y. Lu, and Z. Chen. Web-page summarization using clickthrough data. In *27th SIGIR*, pages 194–201, 2005.

[13] X. Wang, J.-T. Sun, Z. Chen, and C. Zhai. Latent semantic analysis for multiple-type interrelated data objects. In *28th SIGIR*, pages 236–243, 2006.

[14] H. Zha. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *25th SIGIR*, pages 113–120, 2002.